Technical Report: Energy Evaluation of Preamble Sampling MAC Protocols for Wireless Sensor Networks

Giorgio Corbellini, Cédric Abgrall, Emilio Calvanese Strinati, Andrzej Duda

Technical Report: Energy Evaluation of Preamble Sampling MAC Protocols for Wireless Sensor Networks

Giorgio Corbellini*†, Cédric Abgrall*, Emilio Calvanese Strinati*, and Andrzej Duda† *CEA-LETI, MINATEC, Grenoble, France †Grenoble Institute of Technology, CNRS Grenoble Informatics Laboratory UMR 5217, France
Email: [Giorgio.Corbellini, Cedric.Abgrall, Emilio.Calvanese-Strinati]@cea.fr, Andrzej.Duda@imag.fr

Abstract

The technical report presents a simple probabilistic analysis of the energy consumption in preamble sampling MAC protocols. We validate the analytical results with simulations. We compare the classical MAC protocols (B-MAC and X-MAC) with LA-MAC, a method proposed in a companion paper. Our analysis highlights the energy savings achievable with LA-MAC with respect to B-MAC and X-MAC. It also shows that LA-MAC provides the best performance in the considered case of high density networks under traffic congestion.

I. Introduction

Wireless Sensor Networks (WSN) have recently evolved to support diverse applications in various and ubiquitous scenarios, especially in the context of Machine-to-Machine (M2M) networks [1]. Energy consumption is still the main design goal along with providing sufficient performance support for target applications. Medium Access Control (MAC) methods play the key role in saving energy [2] because of the part taken by the radio in the overall energy budget. Thus, the main goal in designing an access method consists of reducing the effects of both *idle listening* during which a device consumes energy while waiting for an eventual transmission and *overhearing* when it receives a frame sent to another device [2].

To save energy, devices aim at achieving low duty cycles: they alternate long sleeping periods (radio switched off) and short active ones (radio switched on). As a result, the challenge of MAC design is to synchronize the instants of the receiver wake-up with possible transmissions of some devices so that the network achieves a very low duty cycle. The existing MAC methods basically use two approaches. The first one synchronizes devices on a common sleep/wake-up schedule by exchanging synchronization messages (SMAC [3], TMAC [4]) or defines a synchronized network wide TDMA structure (LMAC [5], D-MAC [6], TRAMA [7]). With the second approach, each device transmits before each data frame a *preamble* long enough to ensure that intended receivers wake up to catch its frame (Aloha with Preamble Sampling [8], Cycled Receiver [9], LPL (Low Power Listening) in B-MAC [10], B-MAC+ [11], CSMA-MPS [12] aka X-MAC [13], BOX-MAC [14], and DA-MAC [15]). Both approaches converge to the same scheme, called *synchronous preamble sampling*, that uses very short preambles and requires tight synchronization between devices (WiseMAC [16], Scheduled Channel Polling (SCP) [17]).

Thanks to its lack of explicit synchronization, the second approach based on *preamble sampling* appears to be more easily applicable and more scalable than the first synchronous approach. Even if methods based on *preamble sampling* are collision prone, they have attracted great research interest, so that during last years many protocols have been published. In a companion paper, we have proposed LA-MAC, a Low-Latency Asynchronous MAC protocol [18] based on preamble sampling and designed for efficient adaptation of device behaviour to varying network conditions.

In this report, we analytically and numerically compare B-MAC [10], X-MAC [13], and LA-MAC in terms of energy consumption. The novelty of our analysis lies in how we relate the energy consumption to traffic load. In prior energy analyses, authors based the energy consumption on the average Traffic Generation Rate (TGR) of devices [17] as well as on the probability of receiving a packet in a given interval [13]. In contrast to these approaches, which only focus on the consumption of a "transmitter-receiver" couple, we rather consider the global energy cost of a group of neighbour contending devices. Our analysis includes the cost of all radio operations involved in the transmission of data messages, namely the cost of transmitting, receiving, idle listening, overhearing and sleeping.

The motivation for our approach comes from the fact that in complex, dense, and multi-hop networks, traffic distribution is not uniformly spread over the network. Thus, the energy consumption depends on traffic pattern, *e.g. convergecast*, *broadcast*, or *multicast*, because instantaneous traffic load may differ over the network. In our approach, we estimate the energy consumption that depends on the instantaneous traffic load in a given localized area. As a result, our analysis estimates the energy consumption independently of the traffic pattern.

II. Background

We propose to evaluate the energy consumption of a group of contending sensor nodes under three different preamble sampling MAC protocols: B-MAC, X-MAC, and LA-MAC. In complex, dense, and multi-hop networks, the instantaneous traffic distribution over the network is not uniformly spread. For example, in the case of networks with the *convergecast* traffic pattern (all messages go to one sink), the traffic load is higher at nodes that are closer to the sink in terms of number of hops. Due to this *funnelling effect* [19], devices close to the sink exhaust their energy much faster than the others.

The evaluation of the energy consumption in wireless sensor networks is difficult and the energy analyses published in the literature often base the energy consumption of a given protocol on the traffic generation rate of the network [17]. In our opinion, this approach does not fully reflect the complexity of the problem, so we propose to analyze the energy consumption with respect to the number of messages that are buffered in a given geographical area. This approach can represent different congestion situations by varying the instantaneous size of the buffer.

In our analysis, we consider a "star" network composed of a single receiving device (*sink*) and a group of N devices that may have data to send. All devices are within 1-hop radio coverage of each other. We assume that all transmitting devices share a global message buffer for which B sets the number of queued messages, B is then related to network congestion. Among all N devices, N_s of them have at least one packet to send; those nodes with the receiver are called *active* devices. Remaining devices have empty buffers and do not participate in the contention, nevertheless, they are prone to the *overhearing effect*. Thus, there are $N_o = N - N_s$ *overhearers*. According to the global buffer state B, there are several combinations of how to distribute B packets among N sending devices: depending on the number of packets inside the local buffers of active devices, N_s and N_o may vary for each combination. For instance, there can be B active devices with each one packet to send or less than B active devices with some of them having more than one buffered packet.

In the remainder, we explicitly separate the energy cost due to transmission E_t, reception E_r, polling (listening for any radio activity in the channel) E_l, and sleeping E_s. E_o is the overall energy consumption of all overhearers. The overall energy consumption E is the sum of all these energies. The power consumption of respective radio states is P_t, P_r, P_l, and P_s for transmission, reception, channel polling, and sleeping. The power depends on a specific radio device. We distinguish the polling state from the reception state. When a node is performing channel polling, it listens to any channel for activity—to be detected, a radio transmission must start after the beginning of channel polling. Once a radio activity is detected, the device immediately switches its radio state from polling to receiving. Otherwise, the device that is polling the channel cannot change its radio state. The duration of a message over the air is t_d. The time between two wakeup instants is called a *frame* and lapses $t_f = t_l + t_s$, where t_l and t_s are respectively the channel polling duration and the sleep period. These values are related to the duty cycle.

III. Preamble Sampling MAC Protocols

In this section, we provide the details of the analyzed preamble sampling protocols. Figure 1 presents the operation of all protocols.

A. B-MAC

In B-MAC [10], all nodes periodically repeat the same cycle during their lifetime: wake up, listen to the channel, and then go back to sleep. When an active node wants to transmit a data frame, it first transmits a preamble long enough to cover the entire sleep period of a potential receiver. After the preamble the sender immediately transmits the data frame. When the receiver wakes up and detects the preamble, it switches its radio to the receiving mode and listens to the channel until the complete reception of the data frame. Even if the lack of synchronization results in low overhead, the method presents several drawbacks due to the length of the preamble: high energy consumption of transmitters, high latency, and limited throughput. We denote by t_p^B the duration of the B-MAC preamble.

B. X-MAC

In CSMA-MPS [12] and X-MAC [13], nodes periodically alternate sleep and polling periods. After the end of a polling period, each active node transmits a series of short preambles spaced with gaps. During a gap, the transmitter switches to the idle mode and expects to receive an ACK from the receiver. When a receiver wakes up and receives a preamble, it sends an ACK back to the transmitter to stop the series of preambles, which reduces the energy spent by the transmitter. After the reception of the ACK, the transmitter sends a data frame and goes back to sleep. After data reception, the receiver remains awake for a possible transmission of a single additional data frame. If another active node receives a preamble destined to the same receiver it wishes to send to, it stops transmitting to listen to the channel for an incoming ACK. When it overhears the ACK, it sets a random back-off timer at which it will send its data frame. The transmission of a data frame after the back-off is not preceded by any preamble. Note however that nodes that periodically wake up to sample the channel need to keep listening for a duration that is larger than the gap between short preambles to be able to decide whether there is an ongoing transmission or not. The duration of each short preamble is t_p^X and the ACK duration is t_a^X.

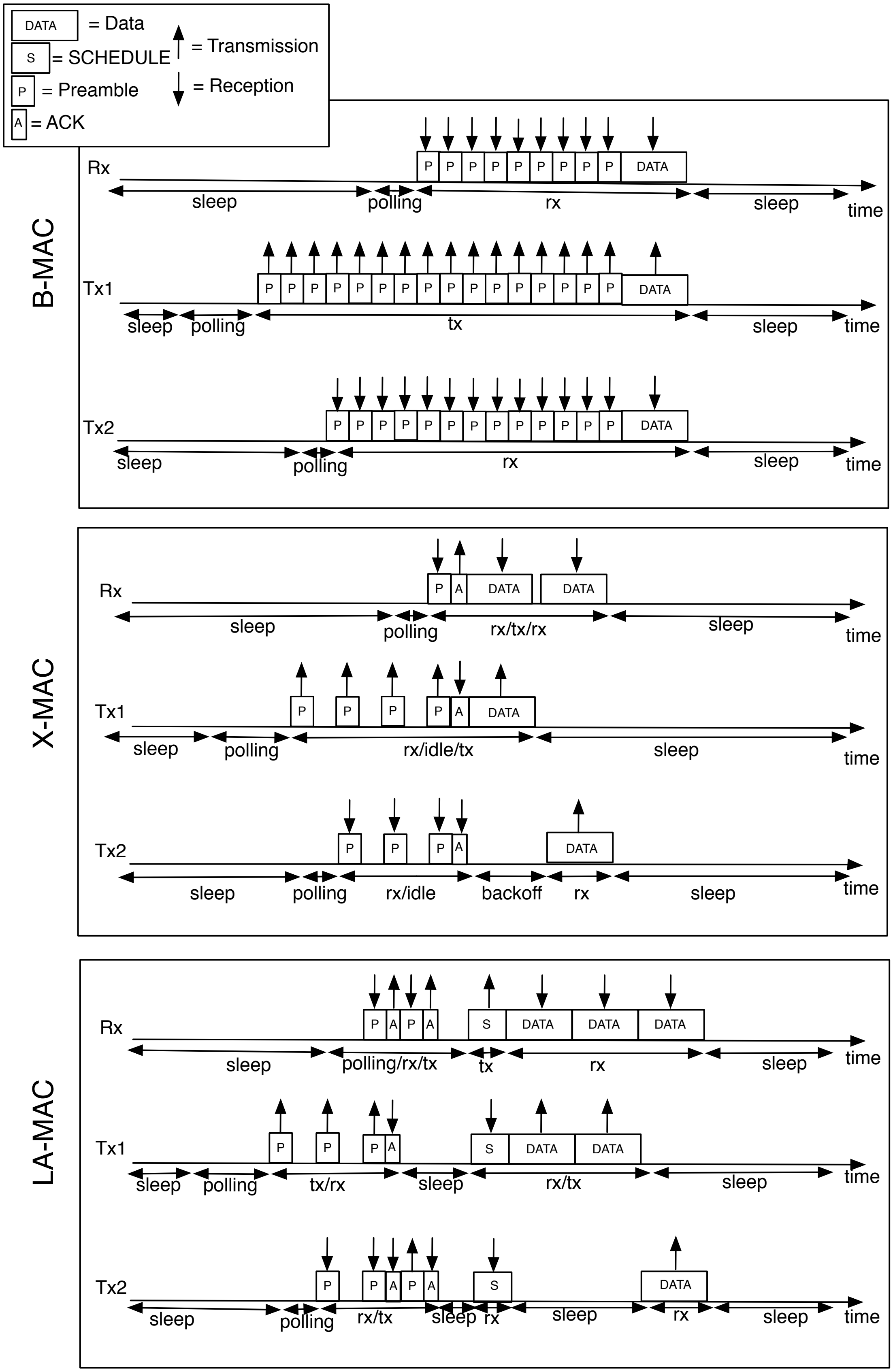

Figure 1. Comparison of analyzed MAC methods.

C. LA-MAC

LA-MAC [18] is a scalable protocol that aims at achieving low latency and limited energy consumption by building on three main ideas: efficient forwarding based on proper scheduling of children nodes that want to transmit, transmissions of frame bursts, and traffic differentiation. It assumes that the network is organized according to some complex structure (tree, DAG, partial mesh) and takes advantage of the network structure to support efficient multi-hop forwarding—a parent of some nodes becomes a coordinator that schedules transmissions in a localized region.

The method periodically adapts local organization of channel access depending on network dynamics such as the number of active users and the instantaneous traffic load. In LA-MAC, nodes periodically alternate long sleep periods and short polling phases. During polling phases each receiver can collect several requests for transmissions included inside short preambles. After the end of its polling period, the node that has collected some preambles processes the requests, compares the priority of requests with the locally backlogged messages and broadcasts a SCHEDULE message. The goal of the SCHEDULE message is to temporarily organize the transmission of neighbor nodes to avoid collisions. If the node that ends its polling has not detected any channel activity and has some backlogged data to send, it starts sending a sequence of short unicast preambles containing the information about the burst to send. As in B-MAC and X-MAC, the strobed sequence is long enough to *wakeup* the receiver. When a receiver wakes up and detects a preamble, it clears it with an ACK frame containing the instant of a *rendezvous* at which it will broadcast the SCHEDULE message. If a second active node overhears a preamble destined to the same destination it wants to send to, it waits for an incoming ACK. After ACK reception, a sender goes to sleep and wakes up at the instant of the *rendezvous*. In Figure 1, we see that after the transmission of an ACK to Tx_1 , Rx device is again ready for receiving preambles from other devices. So, Tx_2 transmits a preamble and receives an ACK during the same *rendezvous*. Preamble clearing continues until the end of the channel polling interval of the receiver.

IV. Energy Analysis

We focus on evaluating energy consumption of a network of *N* transmitters and one receiver. We provide an analytical evaluation of the energy consumption for B-MAC, X-MAC, and LA-MAC based on the instantaneous state of a *global message buffer* B, the number of buffered messages that must be sent by a group of contending devices. In the analysis, we focus our attention on the fact that in a complex sensor networks traffic congestion is not uniformly distributed over the network. For this reason we distinguish and analyse different cases of instantaneous buffer states. We explicit the analytical expressions of energy consumption*E(B)* starting from the case of empty buffers *B=0*, until the generalized expression for large values of B.

A. Global buffer is empty ($B = 0$)

If $B = 0$, all protocols behave in the same way: nodes periodically wakeup, poll the channel for t_l seconds, then go back to sleep because of the absence of channel activity. Thus, the consumption only depends on the time spent in the polling and sleeping states.

$$E^{ALL}(0) = (N + 1) \cdot (t_l \cdot P_l + t_s \cdot P_s) \tag{1}$$

B. Global buffer contains one message ($B = 1$)

If there is one message to send, there are only two devices that are active: the one which has a message in the buffer (N_s = 1) and the destination. Other devices ($N_o = N - 1$) have empty buffers, therefore their energy consumption depends on the channel activity that they can overhear.

B-MAC ($B = 1$)

When message sender wakes up, it polls the channel and then starts sending a long preamble that anticipates data transmission. Even if data is unicast, destination field is not included in preambles; therefore, all nodes need to hear both preamble and the header of the following data in order to know the identity of the intended destination. Provided that devices are not synchronized, each one will hear in average half the duration of a preamble. The cost of transmission is the cost of an entire preamble plus the cost of a data message.

$$E_t^B(1) = (t_p^B + t_d) \cdot P_t \tag{2}$$

The cost of reception is the cost of receiving half of the duration of a preamble plus the cost of receiving data. In packetized radios, a large preamble is obtained by a sequence of short preambles sent one right after the other. For this reason, if a generic device A_2 wakes up and polls the channel while a generic device A_1 is sending a long peamble, radio state of device A_2 will remain in polling state for a short time until the beginning of the next small packet of the large preamble; afterwards the radio will switch in receiving mode consuming more energy. When the receiver (that is not explicitly synchronized with the sender) wakes up, it polls the channel for some activity. Because of lack of synchronization, it may happen that at the time when the receiver wakes up, the sender is performing channel polling. Probability of this event is $p = t_l/t_f$, so if the receiver wakes up during this period, it will perform half of the polling process and then it will listen for the entire preamble. Otherwise, if the receiver wakes up after the end of the polling process of the sender, it will listen for half of the preamble (probability $1-p$). In the remainder of this document we say that with probability p transmitter and receiver are quasi-synchronized.

$$E_r^B(1) = (p \cdot t_p^B + (1-p) \cdot \frac{t_p^B}{2} + t_d) \cdot P_r \tag{3}$$

In addition to the entire polling period of the sender we must consider half of the polling period that must be performed by the receiver weighted by the probability p. The cost of polling process is:

$$E_l^B(1) = (1 + \frac{p}{2}) \cdot t_l \cdot P_l \tag{4}$$

The cost of sleeping activity for the couple transmitter-receiver depends on the time that they do not spend in polling, receiving or transmitting messages.

$$E_s^B(1) = (2 \cdot t_f - (\frac{t_p^B}{2} \cdot (p+3) + 2 \cdot t_d + t_l \cdot (1 + \frac{p}{2}))) \cdot P_s \tag{5}$$

With B-MAC there is not difference in terms of energy consumption between overhearing and receiving a message. Therefore, the cost of overhearing is:

$$E_o^B(1) = N_o \cdot (E_r^B(1) + p \cdot \frac{t_l}{2} \cdot P_l + (t_f - (p \cdot (\frac{t_l}{2} + t_p^B) + (1-p) \cdot \frac{t_p^B}{2} + t_d)) \cdot P_s) \tag{6}$$

X-MAC ($B = 1$)

When the sender wakes up, it polls the channel and starts sending a sequence of unicast preambles separated by a gap for *early* ACK reception. Once the destination has received a preamble, it clears it with an ACK. At this time the sender can transmit its message. After data reception, the receiver remains in polling state for an extra backoff time t_b that is used to receive other possible messages [13]. All devices that have no messages to send and that overhear channel activity go to sleep.

The expected number of preambles that are needed to *wakeup* the receiver is γ^X. The average number of preambles depends on the duration of polling period, preamble and ACK messages as well as the duration of an entire frame [13]. If the couple sender-receiver is not synchronized, each preamble of the sender has the same probability to be heard or not by the receiver. So we can consider each preamble transmission as a trial of a geometric distribution with probability $1/\gamma^X$. The expected value γ^X is the inverse of the *collision probability* of one preamble over the polling period of the receiver. We remark that before there is a collision between preamble and polling period there are $(\gamma^X - 1)$ preambles whose energy is wasted.

$$\gamma^X = (\frac{t_l - t_a^X - t_p^X}{t_f})^{-1} \tag{7}$$

Total amount of energy that is due to the activity of transmitting one message depends on the average number of preambles that must be sent (γ^X) and the cost of *early* ACK reception. Provided that wakeup schedules of nodes are not synchronous, it may happen that when the receiver wakes up, the sender is already performing channel polling (transmitter and receiver are quasi-synchronized with probability p).

In the case of quasi-synchronization, the receiver will perform in average half of the polling process and afterwards it the will be able to clear the very first preamble of the sequence. In this case, the cost of transmission only includes the transmission of one preamble and the cost of receiving the ACK. Otherwise, if nodes are not synchronous (the receiver wakes up after the end of the polling process of the sender), the receiver will cause the sender to waste energy for the transmission of γ^X preambles and the wait for an ACK (we consider waiting for ACK as a polling state) before it can hear one of them. The energy consumption of all activities of polling is reported separately in $E_l^X(1)$. Transmission cost is:

$$\begin{aligned} E_t^X(1) = \quad & (1-p) \cdot \gamma^X \cdot t_p^X \cdot P_t + p \cdot t_p^X \cdot P_t + t_a^X \cdot P_r + t_d \cdot P_t \\ = \quad & ((1-p) \cdot \gamma^X + p) \cdot t_p^X \cdot P_t + t_a^X \cdot P_r + t_d \cdot P_t \end{aligned} \tag{8}$$

The cost of the receiving activity does not depend on p and is represented by the transmission of one ACK plus the reception of both data and preamble.

$$E_r^X(1) = (t_d + t_p^X) \cdot P_r + t_a^X \cdot P_t \tag{9}$$

With probability $1-p$ (no synchronization) the receiver will wakeup while the sender is already transmitting a preamble (or it is waiting for an *early* ACK). Otherwise (with probability p) the receiver will perform in average, only half of its polling period.

More in details, if the active couple is quasi-synchronized they simultaneously perform channel sensing, then the sender starts the preamble transmission. As far as the sender is concerned, we must consider both the entire polling period and the time that the sender waits for *early* ACK without any answer (event that happens with probability $1-p$).

$$\begin{aligned} E_l^X(1) = &\quad ((t_l + (1-p)\cdot(\gamma^X - 1)\cdot t_a^X) + ((1-p)\cdot\frac{t_p^X + t_a^X}{2} + p\cdot\frac{t_l}{2}) + t_b)\cdot P_l \\ = &\quad ((1-p)\cdot(\frac{t_p^X + t_a^X}{2} + (\gamma^X - 1)\cdot t_a^X) + (\frac{p}{2}+1)\cdot t_l + t_b)\cdot P_l \end{aligned} \quad (10)$$

Sleep activity of the active couple is twice a frame duration minus the time that both devices are active.

$$\begin{aligned} E_s^X(1) = &\quad (2\cdot t_f - (t_l + ((1-p)\cdot\gamma^X + p)\cdot(t_p^X + t_a^X) + t_d) - (p\cdot\frac{t_l}{2} + t_p^X + t_a^X + (1-p)\cdot\frac{t_p^X + t_a^X}{2} + t_d + t_b))\cdot P_s \\ = &\quad (2\cdot t_f - 2\cdot t_d - p\cdot\frac{t_l}{2} - t_p^X - t_a^X - (1-p)\cdot\frac{t_p^X + t_a^X}{2} - t_l - ((1-p)\cdot\gamma^X + p)\cdot(t_p^X + t_a^X) - t_b)\cdot P_s \end{aligned} \quad (11)$$

As other devices, the overhearers can wakeup at a random instant. However, differently from active agents, as soon as they overhear some activity they go back to sleep. Therefore, their energy consumption depends on the probability that such nodes wake up while the channel is busy or not. The probability that at wakeup instant the channel is free depends on polling duration, buffer states, number of senders etc. In Figure 2 we show all the possible situations that can happen. In the Figure we consider as reference instant, the time at which the transmitter wakes up (root of the tree). With probability p, the transmitter (Tx) and the Receiver (Rx) are quasi-synchronized, not synchronized otherwise (probability $(1-p)$). With probability $p \cdot p$ both the receiver and a generic overhearer are quasi-synchronized with the transmitter, this is the Case 1 in the tree. In the remainder we explicit the expressions for all possible combination of wakeup of the overhearers. Probabilities are specified in Figure 2.

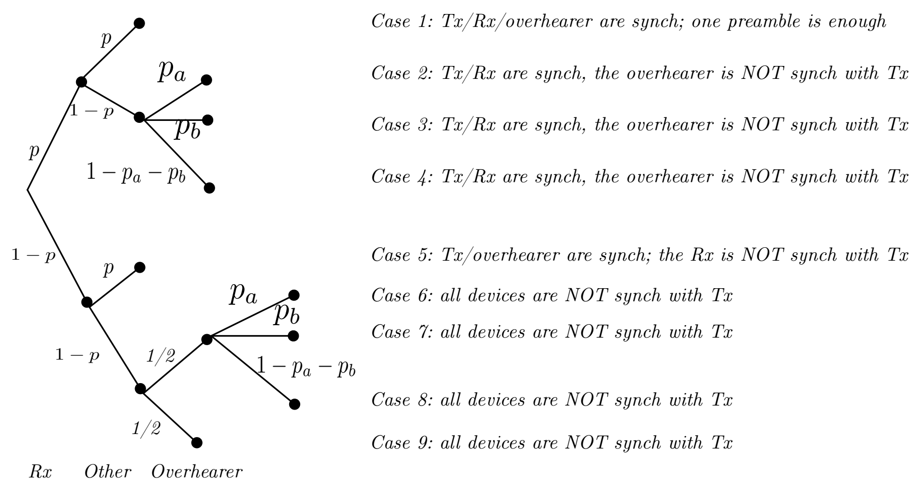

Figure 2. X-MAC. Tree of different combinations for wakeup schedules.

- Case 1: Sender, receiver and overhearer are quasi-synchronized. The overhearer will sense a preamble that is not intended to it, therefore it goes back to sleep.

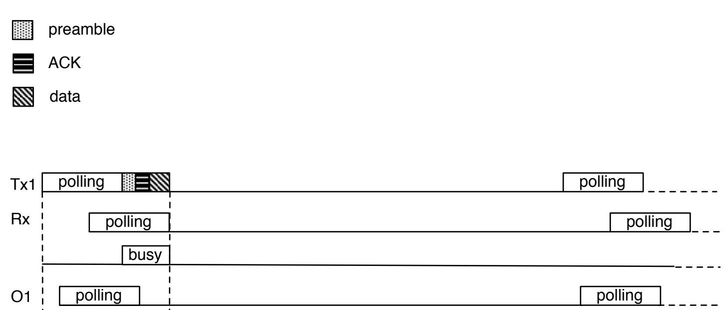

Figure 3. X-MAC protocol, global buffer size $B = 1$: Overhearing situations for Case 1.

$$E^X_{Case_1,o} = \frac{t_l}{2} \cdot P_l + t^X_p \cdot P_r + (t_f - \frac{t_l}{2} - t^X_p) \cdot P_s \tag{12}$$

- Case 2, 3, 4: The receiver is synchronized with the Sender but not the overhearer. When the overhearer wakes up, it can overhear different messages (preamble, ACK or data) as well as clear channel. Possible situations are summarized in Figure 4.

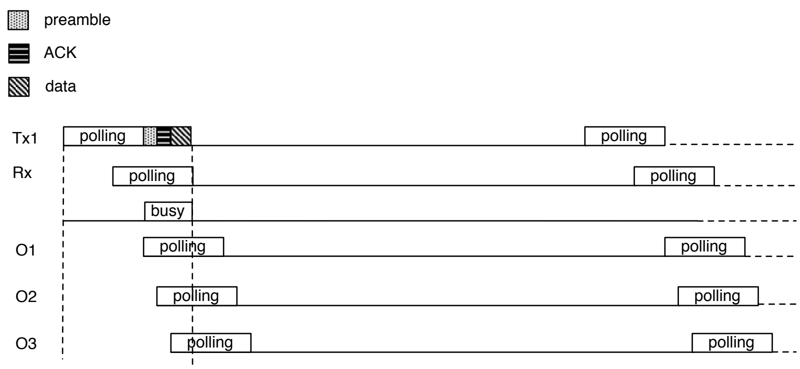

Figure 4. X-MAC protocol, global buffer size $B = 1$: Overhearing situations for Cases 2, 3 and 4.

- Case 2: If the ovehearer wakes up during a preamble transmission, it will ovehear the following ACK and afterwards go back to sleep. The probability for the overhearer to wakeup during a preamble is $p_a = t^X_p / t_f$.

$$E^X_{Case_2,o} = \frac{t^X_p}{2} \cdot P_l + t^X_a \cdot P_r + (t_f - \frac{t^X_p}{2} - t^X_a) \cdot P_s \tag{13}$$

- Case 3: If the ovehearer wakes up during an ACK transmission, it will listen to the following data message and afterwards go back to sleep. The probability for the overhearer to wakeup during an ACK is $p_b = t^X_a / t_f$.

$$E^X_{Case_3,o} = \frac{t^X_a}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t^X_a}{2} - t_d) \cdot P_s \tag{14}$$

- Case 4: The ovehearer will either wake up during data transmission or during the silent period that follows. In both cases when the sender wakes up and senses the channel, it claims a free channel because in the first case any message is received and in the latter case nobody is transmitting anything. Therefore, the overhearer performs an entire polling process and goes back to sleep immediately after. The probability for this event to happen is $1 - p_a - p_b$.

$$E^X_{Case_4,o} = t_l \cdot P_l + (t_f - t_l) \cdot P_s \tag{15}$$

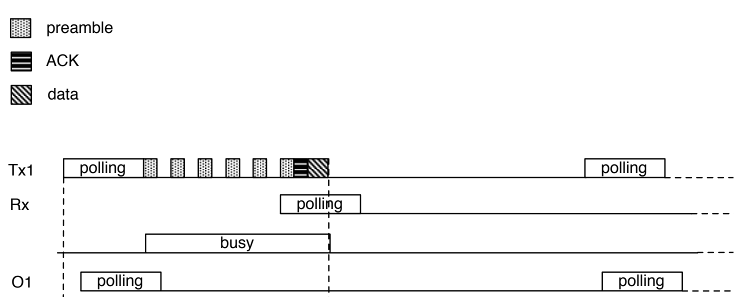

Figure 5. X-MAC protocol, global buffer size $B = 1$: Overhearing situations for Case 5.

- Case 5: Similarly to Case 1, if the overhearer is quasi-synchronized with the transmitter it will overhear the first preamble even if the receiver is still sleeping. The energy cost is:

$$E^X_{Case_5,o} = E^X_{Case_1,o} \tag{16}$$

- Cases 6, 7, 8: If neither the receiver nor the overhearer are synchronized with the sender, it may happen that the receiver wakes up before the overhearer. Therefore, similarly to cases 2, 3 and 4 we have different situations. Cases 6, 7, 8 are respectively similar to 2, 3 and 4:

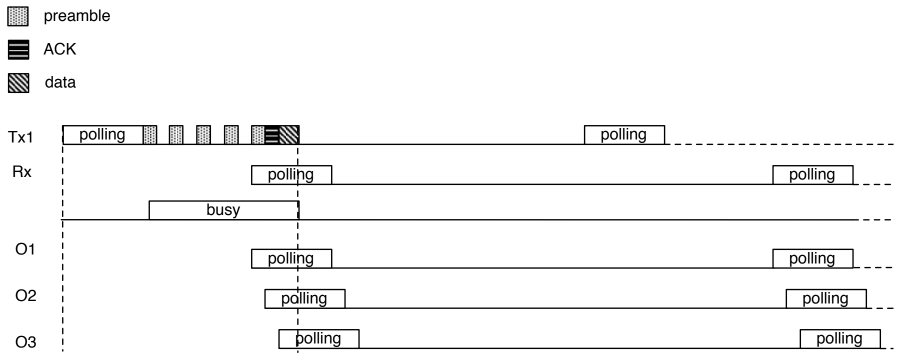

Figure 6. X-MAC protocol, global buffer size $B = 1$: Overhearing situations for Cases 6, 7 and 8.

$$E^X_{Case_6,o} = E^X_{Case_2,o} \tag{17}$$

$$E^X_{Case_7,o} = E^X_{Case_3,o} \tag{18}$$

$$E^X_{Case_8,o} = E^X_{Case_4,o} \tag{19}$$

- Case 9: If the overhearer wakes up before the intended destination, it will receive a preamble and go back to sleep. The cost in this case is:

$$E^X_{Case_9,o} = t^X_p \cdot P_r + \frac{t^X_p + t^X_a}{2} \cdot P_l + (t_f - \frac{t^X_p + t^X_a}{2} - t^X_p) \cdot P_s \tag{20}$$

The overall energy cost is the sum of the costs of each case weighted by the probability of the given case to happen

$$E^X_o(1) = N_o \cdot \sum_{i=1}^{9} p_{Case_i} \cdot E^X_{Case_i,o} \tag{21}$$

LA-MAC ($B = 1$)

In LA-MAC, wakeup schedules are assumed randomly distributed. When sender wakes up polls channel and then it sends preambles as in X-MAC. However, differently from X-MAC after *early* ACK reception, the sender goes back to sleep and waits for SCHEDULE message to be sent. When the intended destination receives one preamble, it clears it with an *early* ACK and completes its polling period in order to detect other possible preambles to clear. Immediately after the end of polling period, the receiver processes the requests and broadcasts a SCHEDULE message. In LA-MAC, overhearers go to sleep as soon as they receive any unicast message (preamble, ACK or data) as well as the SCHEDULE (that is a broadcast message).

Due to lack of synchronization, expected number of preambles follows X-MAC expression with different size of preambles t^L_p and ACK t^L_a. More in detail, when the sender wakes up, it performs an entire channel polling process before starting transmitting strobed preambles. When the receiver wakes up, it polls the channel. With probability $p = t_l/t_f$ the sender and receiver are quasi-synchronized; so with probability p the sender is still polling the channel when the receiver wakes up.

When the sender wakes up, it polls the channel and starts sending preambles to *wakeup* the receiver. With probability p, the first preamble that is sent will wake up the receiver and the sender will immediately receive an *early* ACK. Otherwise, if nodes are not synchronized (probability $(1-p)$) the sender will wake up its destination in average after γ^L preambles. $E^L_t(1)$ is similar to the cost of X-MAC plus the cost of receiving the SCHEDULE:

$$\begin{aligned} E^L_t(1) = & \quad (1-p) \cdot \gamma^L \cdot t^L_p \cdot P_t + p \cdot t^L_p \cdot P_t + t^L_a \cdot P_r + t_d \cdot P_t + t_g \cdot P_r \\ = & \quad ((1-p) \cdot \gamma^L + p) \cdot t^L_p \cdot P_t + (t^L_a + t_g) \cdot P_r + t_d \cdot P_t \end{aligned} \tag{22}$$

Cost of reception depends on the duration of a preamble, an ACK, a data and a SCHEDULE message.

$$E^L_r(1) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t \tag{23}$$

When the sender wakes up, it performs a full polling period before the beginning of the strobed preambles. Moreover, the degree of synchronization between active nodes also influences the consumption. If active nodes are not synchronized, the sender will poll the channel $(\gamma^L - 1)$ times in order to wait for *early* ACK. Differently from X-MAC, the receiver will complete its polling period even if it clears one preamble, so its radio will remain in polling state for the duration of a full polling period less the time for preamble reception and ACK transmission.

$$E_l^L(1) = ((t_l + (1-p) \cdot (\gamma^L - 1) \cdot t_a^L) + (t_l - t_p^L - t_a^L)) \cdot P_l \tag{24}$$

When the active nodes are not transmitting, receiving or polling the channel they can sleep.

$$E_s^L(1) = (2 \cdot t_f - (t_l + (1-p) \cdot \gamma^L \cdot t_p^L + p \cdot t_p^L + t_a^L + (1-p) \cdot (\gamma^L - 1) \cdot t_a^L + t_d + t_g) - (t_l + t_d + t_g)) \cdot P_s \tag{25}$$

As in X-MAC as soon as overhearers receive a message they go back to sleep. Therefore their energy consumption depends on the probability that such nodes wake up while the channel is busy or not. All the possible combinations of wakeup schedules with relative probabilities are shown in Figure 7 .

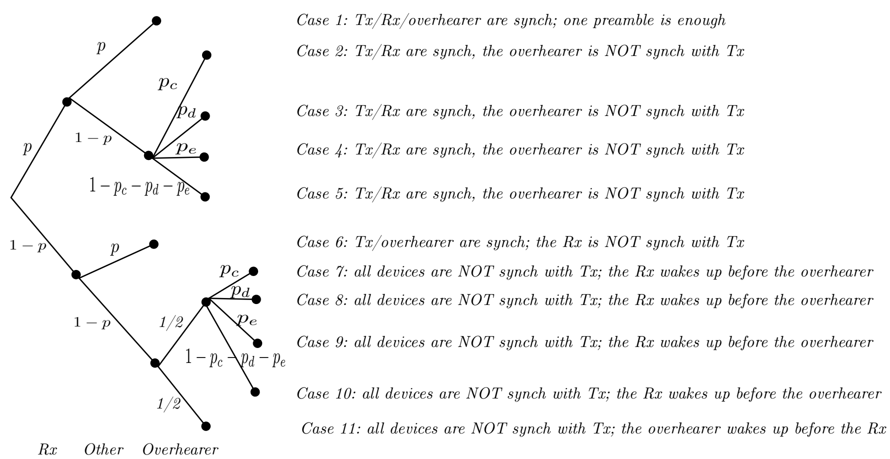

Figure 7. Lamac. Tree of different wakeup cases.

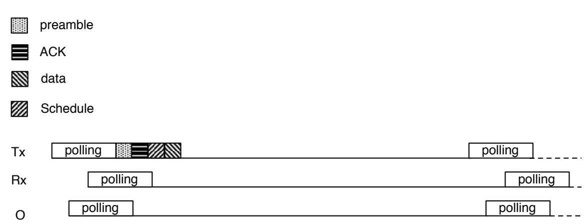

Figure 8. LA-MAC protocol, global buffer size $B = 1$: Overhearing situations for Case 1.

- Case 1: Sender, receiver and overhearer are quasi-synchronized. The overhearer will sense a preamble that is not intended to it and goes back to sleep. Probability of this event is $p \cdot p$:

$$E^L_{Case_1,o} = \frac{t_l}{2} \cdot P_l + t_p^L \cdot P_r + (t_f - \frac{t_l}{2} - t_p^L) \cdot P_s \tag{26}$$

- Case 2, 3, 4, 5: the receiver is synchronized with the sender. Nevertheless, the overhearer is not synchronized with the sender. When the overhearer wakes up, it can receive different messages (preamble, ACK, SCHEDULE or data) as well as clear channel.
  - Case 2: If the ovehearer wakes up during a preamble transmission, it will receive in average half of a preamble and overhear the following ACK. Afterwards it will go back to sleep. Probability of this event is $p \cdot (1-p) \cdot p_c$, where

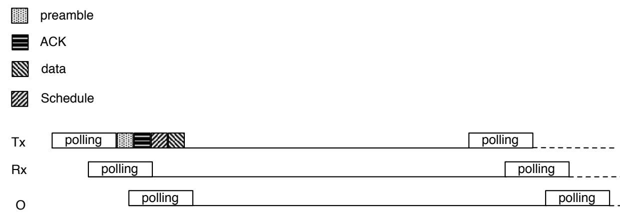

Figure 9. LA-MAC protocol, global buffer size $B = 1$: Overhearing situations for Cases 2, 3, 4 and 5.

$p_c = t_p^L / t_f$ represents the event that wakeup instant of the overhearer is slightly after the end of polling process of the sender:

$$E^L_{Case_2,o} = \frac{t_p^L}{2} \cdot P_l + t_a^L \cdot P_r + (t_f - \frac{t_p^L}{2} - t_a^L) \cdot P_s \tag{27}$$

- Case 3: If the ovehearer wakes up during an ACK transmission, it will sense a silent period and ovehear the following SCHEDULE message. Afterwards it will go back to sleep. Probability of this event is $p \cdot (1-p) \cdot p_d$, where $p_d = t_a^L / t_f$ includes the event that wakeup instant of the overhearer happens at least after the transmission of a preamble. p_d neglects the time that elapses between the end of the ACK and the end of channel polling of the receiver. In other words, with p_d we assume that schedule message is sent immediately after the transmission of ACK:

$$E^L_{Case_3,o} = \frac{t_a^L}{2} \cdot P_l + t_g \cdot P_r + (t_f - \frac{t_a^L}{2} - t_g) \cdot P_s \tag{28}$$

- Case 4: If the overhearer wakes up during the transmission of a SCHEDULE, it will hear the following data and then go to sleep. Probability of this event is $p \cdot (1-p) \cdot p_e$, with $p_e = t_g / t_f$ we assume that the wakeup instant of the overhearer happens in average at the middle of schedule transmission:

$$E^L_{Case_4,o} = \frac{t_g}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_g}{2} - t_d) \cdot P_s \tag{29}$$

- Case 5: The ovehearer will either wake up during data transmission or sense a free channel because both sender and receiver are already sleeping. Therefore the overhearer performs an entire polling and goes back to sleep. Probability of this event is $p \cdot (1-p) \cdot (1 - p_c - p_d - p_e)$:

$$E^L_{Case_5,o} = t_l \cdot P_l + (t_f - t_l) \cdot P_s \tag{30}$$

- Case 6: Similarly to Case 1, if the overhearer is quasi-synchronized with the sender, with probability $(1-p) \cdot p$, the energy cost is:

$$E^L_{Case_6,o} = \frac{t_l}{2} \cdot P_l + t_p^L \cdot P_r + (t_f - \frac{t_l}{2} - t_p^L) \cdot P_s \tag{31}$$

- Cases 7, 8, 9, 10: If neither the receiver nor the ovehearer are synchronized with sender, it may happen that the receiver wakes up before the overhearer. We distinguish the situations of quasi-synchronization of the couple overhearer-receiver and lack of synchronization. In cases 7 and 8, overhearer is quasi-synchronized with the receiver:

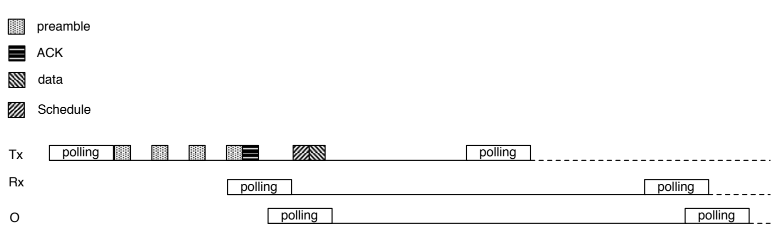

Figure 10. Lamac. Possible wakeup instants of overhearers. Cases 7,8,9, 10.

- Case 7: There is a probability to overhear a preamble. Such a probability is $(1-p) \cdot (1-p) \cdot 1/2 \cdot p_c$. Consumption of this case is the same of Case 2:

$$E^L_{Case_7,o} = E^L_{Case_2,o} \tag{32}$$

– Case 8: There is a probability to overhear an ACK. Such a probability is $(1-p)\cdot(1-p)\cdot 1/2\cdot p_d$. Consumption of this case is the same of Case 3:

$$E^L_{Case_8,o} = E^L_{Case_3,o} \tag{33}$$

If the overhearer and the receiver are not synchronized:

– Case 9: There is a probability to overhear a SCHEDULE. Such a probability is $(1-p)\cdot(1-p)\cdot 1/2\cdot p_e$. Consumption of this case is the same of Case 4:

$$E^L_{Case_9,o} = E^L_{Case_4,o} \tag{34}$$

– Case 10: There is a probability to overhear a data message. Such a probability is $(1-p)\cdot(1-p)\cdot 1/2\cdot(1-p_c-p_d-p_e)$. Consumption of this case is the same of Case 5:

$$E^L_{Case_{10},o} = E^L_{Case_5,o} \tag{35}$$

- Case 11: Otherwise, if the overhearer wakes up before the destination, it will receive one preamble (whichever preamble amongst γ^L) and go back to sleep. The cost in this case is:

$$E^L_{Case_{11},o} = \frac{t^L_p + t^L_a}{2}\cdot P_l + t^L_p \cdot P_r + (t_f - \frac{t^L_p + t^L_a}{2} - t^L_p)\cdot P_s \tag{36}$$

The overall energy cost is the sum of the costs of each case weighted by the probability of the given case to happen:

$$E^L_o(1) = N_o \cdot \sum_{i=1}^{11} p_{Case_i}\cdot E^L_{Case_i,o}. \tag{37}$$

C. Global buffer contains two messages ($B = 2$)

If $B = 2$, there can be either one sender with two waiting messages, or two senders with only one message. The number of overhearers will be $N_o = N - 1$ if there is just one sender, $N_o = N - 2$ otherwise. The probability that two messages are in different buffers is equal to $(N-1)/N$.

B-MAC ($B = 2$)

The overall energy consumption for transmission and reception when $B \geq 1$ is linear with the global number of packets in the buffer, independently on how packets are distributed in the different local buffers, *i.e.,* independently of the number of senders. In fact, due to the long preamble to send ($t^b_p = t_f$), there can be only one sender per frame. Thus, we have the following relation: $E^B(B) = B\cdot E^B(1) = B\cdot(E^B_t(1) + E^B_r(1) + E^B_l(1) + E^B_s(1) + E^B_o(1))$.

Such a relation highlights the limitations of B-MAC protocol, since high-loaded traffic can hardly been addressed.

X-MAC ($B = 2$)

After the reception of the first data message, the receiver remains in polling state for an extra back-off time t_b during which it can receive a second message. The energy consumed for the transmission of the first packet is the same as the energy defined in the previous subsection $E^X_t(1)$; then an additional cost of the transmission for the second message must be considered.

Differently from B-MAC, the distribution of messages in the buffers has impact on protocol behaviour. With probability $1/N$ both packets are in the same buffer; otherwise two different senders are implicated. In case of multiple senders we need to study how wakeup instants of the active agents are scheduled with respect to each others. Wakeup instant of different agents are all independent. We assume that the frame begins at the wakeup instant of the first transmitter; scenarios that may happen are illustrated on Figure 11 with the corresponding happening probability.

- Case 1: There are two senders and a receiver, all quasi-synchronized. The very first preamble sent by the first transmitter is cleared by the receiver who sends an ACK; the second transmitter hears both the preamble and the ACK. Probability of this scenario is $p_{Case_1} = (N-1)/N\cdot p\cdot p$. We have:

$$E^X_{Case_1,t}(2) = t^X_p\cdot P_t + t^X_a\cdot P_r + (t^X_p + t^X_a)\cdot P_r + 2\cdot t_d\cdot P_t \tag{38}$$

$$E^X_{Case_1,r}(2) = (t^X_p + 2\cdot t_d)\cdot P_r + t^X_a\cdot P_t \tag{39}$$

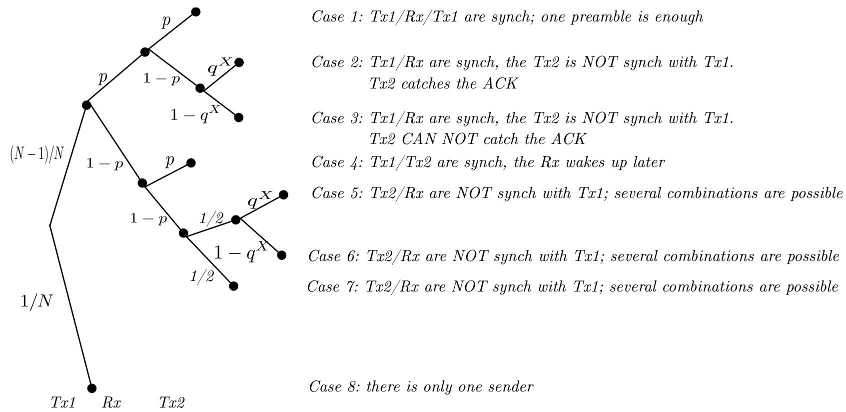

Figure 11. X-MAC protocol: Probability tree of wakeup combinations with global buffer size $B = 2$. There are one sender and one or two transmitters.

$$E^X_{Case_1,l}(2) = (t_l + \frac{t_l}{2} + \frac{t_l}{2}) \cdot P_l \tag{40}$$

$$E^X_{Case_1,s}(2) = (3 \cdot t_f - (t_l + t^X_p + t^X_a + t_d) - (\frac{t_l}{2} + t^X_p + t^X_a + t_d) - (\frac{t_l}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \tag{41}$$

Depending on wakeup instants of overhearers several situations may happen. If the overhearer is quasi-synchronized with one of the three active agents (the receiver or one of the two senders), then it will sense a busy channel (cf. Figure 12). We assume that an overhearer polls the channel for some time and then overhears a message that can be a preamble, an ACK or a data. For simplicity, we assume that the overhearer polls the channel for an average duration of half of a polling period and then it overhears a data (the largest message that can be overheard). Probability to wakeup during a busy period is $p^X_{case1,B=2} = (t^X_p + t^X_a + 2 \cdot t_d)/t_f$. Otherwise, the overhearer wakes up while channel is free, it polls the channel and then goes back to sleep.

$$E^X_{Case_1,o}(2) = N_o \cdot (p^X_{case1,B=2} \cdot (\frac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_l}{2} - t_d) \cdot P_s) + (1 - p^X_{case1,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \tag{42}$$

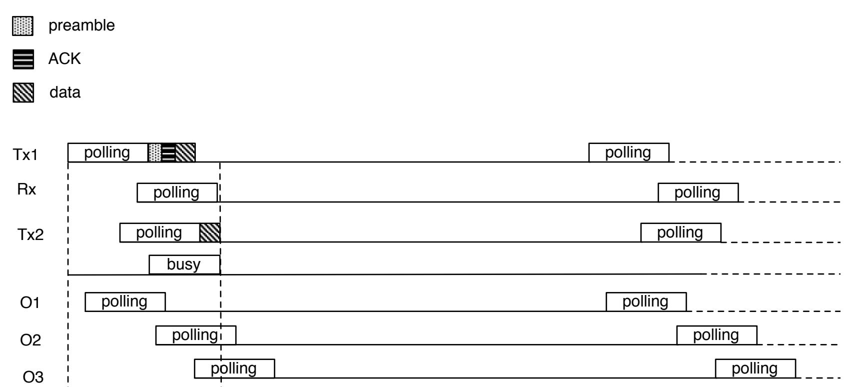

Figure 12. X-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 1.

- Case 2: The first sender and receiver are quasi-synchronized, in contrast to the second sender (cf. Figure 13). The only possibility for the second sender to send data in the current frame is to manage to catch the ACK of the receiver during its polling period. This event happens with probability $q^X = (t_l - t^X_a)/t_f$. Probability of this scenario is $p_{Case_2} = (N-1)/N \cdot p \cdot (1-p) \cdot q^X$.

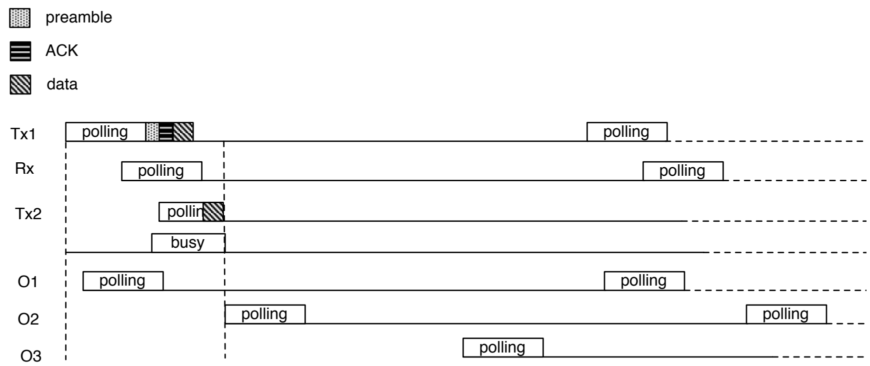

Figure 13. X-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 2.

Energy consumption of this second scenario is quite the same as the one of Case 1 but event probability is different. Since the second sender is not quasi-synchronised, it cannot hear the full preamble sent by the first sender and has a shorter polling period.

$$E^X_{Case_2,t}(2) = E^X_{Case_1,t}(2) - t^X_p \cdot P_r \tag{43}$$

$$E^X_{Case_2,r}(2) = E^X_{Case_1,r}(2) \tag{44}$$

$$E^X_{Case_2,l}(2) = E^X_{Case_1,l}(2) - \frac{t_l - t^X_p}{2} \cdot P_l \tag{45}$$

$$E^X_{Case_2,s}(2) = E^X_{Case_1,s}(2) + \frac{t_l + t^X_p}{2} \cdot P_l \tag{46}$$

We assume that the probability of busy channel is the same as the previous scenario. So, overhearing consumption is unchanged.

$$E^X_{Case_2,o}(2) = E^X_{Case_1,o}(2) \tag{47}$$

- Case 3: With probability $1 - q^X$, the second sender wakes up too late and cannot catch the ACK. In this case, it goes back to sleep and it will transmit its data during the next frame. Energy cost is the sum of the transmission cost for the first packet in the current frame and for the second packet in the following frame. The cost for second frame is the same as $E^X(1)$. This scenario happens with probability $p_{Case_3} = (N-1)/N \cdot p \cdot (1-p) \cdot (1-q^X)$.

$$E^X_{Case_3,t}(2) = t^X_p \cdot P_t + t^X_a \cdot P_r + t_d \cdot P_t + E^X_t(1) \tag{48}$$

$$E^X_{Case_3,r}(2) = t^X_p \cdot P_r + t^X_a \cdot P_t + t_d \cdot P_r + E^X_r(1) \tag{49}$$

$$E^X_{Case_3,l}(2) = (t_l + t_l + \frac{t_l}{2}) \cdot P_l + E^X_l(1) \tag{50}$$

$$E^X_{Case_3,s}(2) = (3 \cdot t_f - (t_l + t^X_p + t^X_a + t_d) - t_l - (\frac{t_l}{2} + t^X_p + t^X_a + t_d)) \cdot P_s + E^X_s(1) \tag{51}$$

In the second frame, the first sender has nothing to send any more and can be counted as an overhearer. Even if the number of overhearers increases from the first frame to the second, energy cost per overhearer remains the same as the case of single message to send ($B = 1$):

$$E^X_{Case_3,o}(2) = (N_o + (N_o + 1)) \cdot \frac{E^X_o(1)}{N_o + 1} \tag{52}$$

- Case 4: First and second senders are quasi-synchronized but the receiver wakes up later. In this scenario, the first sender sends a strobed preamble until the receiver wakes up and sends an ACK; the second sender hears the entire strobed preamble sequence and then sends its data during the back-off time. Between short preambles, senders poll channel waiting for an ACK from receiver. Probability of this scenario is $p_{Case_4} = (N-1)/N \cdot (1-p) \cdot p$.

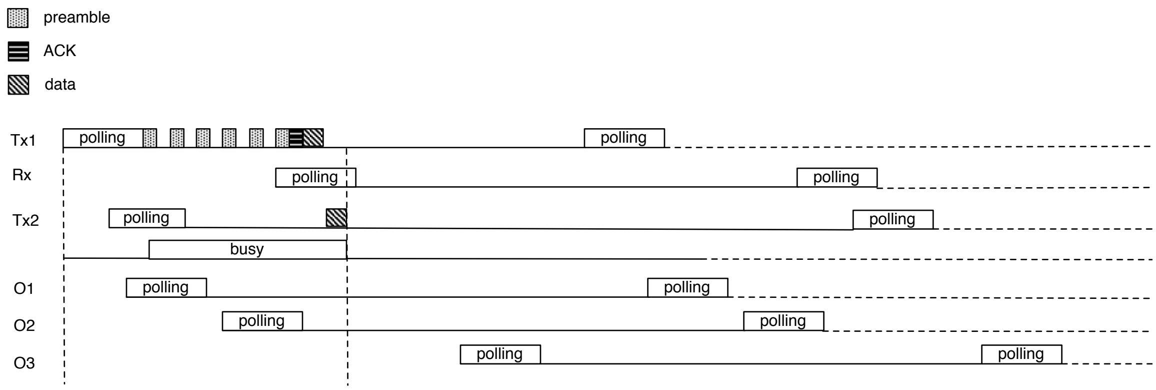

Figure 14. X-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 4.

$$E^X_{Case_4,t}(2) = \gamma^X \cdot t^X_p \cdot (P_t + P_r) + 2 \cdot t^X_a \cdot P_r + 2 \cdot t_d \cdot P_t \tag{53}$$

$$E^X_{Case_4,r}(2) = (t^X_p + 2 \cdot t_d) \cdot P_r + t^X_a \cdot P_t \tag{54}$$

$$E^X_{Case_4,l}(2) = (t_l + \frac{t_l}{2} + 2 \cdot (\gamma^X - 1) \cdot t^X_a + \frac{t^X_p + t^X_a}{2}) \cdot P_l \tag{55}$$

$$\begin{aligned} E^X_{Case_4,s}(2) = \quad & (3 \cdot t_f - (t_l + \gamma^X \cdot (t^X_p + t^X_a) + t_d) - (\tfrac{t_l}{2} + \gamma^X \cdot (t^X_p + t^X_a) + t_d) \\ - \quad & (\tfrac{t^X_p + t^X_a}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{56}$$

When receiver wakes up after than both senders, the probability that an overhearer wakes up during a transmission of a preamble is higher than in previous scenarios. If this happens, the overhearer performs a very short polling, overhears a message (most probably a preamble) and then goes back to sleep. For simplicity we assume that the overhearer will perform half of $(t^X_p + t^X_a)$ of polling process and than overhears an entire preamble. Probability of busy channel is thus $p^X_{case4,B=2} = (\gamma^X \cdot (t^X_p + t^X_a) + 2 \cdot t_d)/t_f$.

$$\begin{aligned} E^X_{Case_4,o}(2) = \quad & N_o \cdot (p^X_{case4,B=2} \cdot (\tfrac{t^X_p + t^X_a}{2} \cdot P_l + t^X_p \cdot P_r + (t_f - \tfrac{t^X_p + t^X_a}{2} - t^X_p) \cdot P_s) \\ + \quad & (1 - p^X_{case4,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \end{aligned} \tag{57}$$

- Cases 5, 6, 7: Second sender and receiver are not synchronized with first sender; the behaviour of the protocol depends on which device among the second sender and the receiver will wake up as first.
  - Case 5: The receiver wakes up as first. Similarly to Case 2, the only possibility for the second transmitter to send data in the current frame is to catch the ACK of the receiver during its polling. This event happens with probability $q^X = (t_l - t^X_a)/t_f$. However, there is also the possibility for Tx_2 to catch the preamble sent by Tx_1 that just precedes the overheard ACK. Such eventuality can happen with probability $u^X = (t^X_p + t^X_a)/(2 \cdot t^X_p + t^X_a)$. This scenario happens with probability $p_{Case_5} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2} \cdot q^X$.

$$E^X_{Case_5,t}(2) = (\gamma^X \cdot t^X_p + t_d) \cdot P_t + t^X_a \cdot P_r + (u^X \cdot t^X_p + t^X_a) \cdot P_r + t_d \cdot P_t \tag{58}$$

$$E^X_{Case_5,r}(2) = (t^X_p + 2 \cdot t_d) \cdot P_r + t^X_a \cdot P_t \tag{59}$$

$$E^X_{Case_5,l}(2) = (t_l + (\gamma^X - 1) \cdot t^X_a + \frac{t^X_p + t^X_a}{2} + u^X \cdot \frac{t^X_p + t^X_a}{2} + (1 - u^X) \cdot \frac{t^X_p}{2}) \cdot P_l \tag{60}$$

$$\begin{aligned} E^X_{Case_5,s}(2) = \quad & (3 \cdot t_f - (t_l + \gamma^X \cdot (t^X_p + t^X_a) + t_d) \\ - \quad & (u^X \cdot \tfrac{t^X_p + t^X_a}{2} + (1 - u^X) \cdot \tfrac{t^X_p}{2} + u^X \cdot t^X_p + t^X_a + t_d) \\ - \quad & (\tfrac{t^X_p + t^X_a}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{61}$$

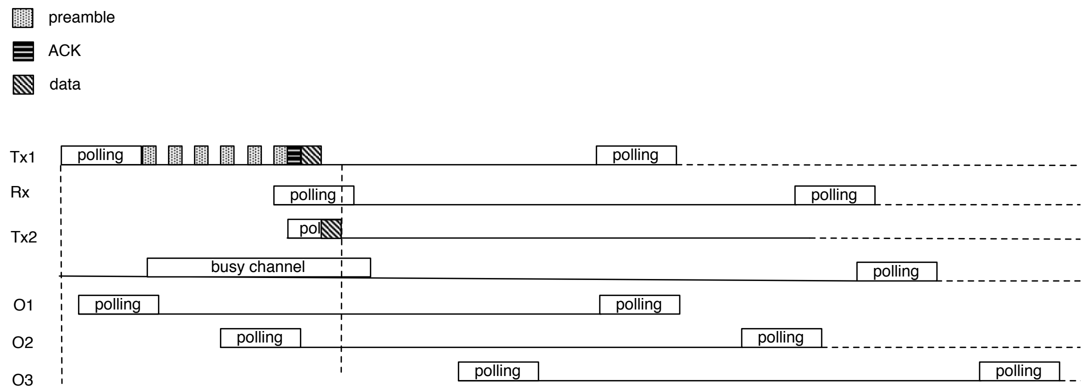

Figure 15. X-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 5.

As in the previous case, the overhearer perceives a very busy channel because of the transmission of preambles; so when it wakes up it will perform half of $(t_p^X + t_a^X)$ in polling state before overhearing an entire preamble. Probability of busy channel is $p_{case5}^X = p_{case4}^X$.

$$E_{Case_5,o}^X(2) = E_{Case_4,o}^X(2) \tag{62}$$

– Case 6: The receiver wakes up as first. Similarly to Case 3, with probability $(1 - q^X)$, the second sender wakes up too late and cannot catch the ACK from the receiver. Thus it goes back to sleep and will transmit its data during the next frame. This scenario happens with probability $p_{Case_6} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2} \cdot (1 - q^X)$.

$$E_{Case_6,t}^X(2) = \gamma^X \cdot t_p^X \cdot P_t + t_a^X \cdot P_r + t_d \cdot P_t + E_t^X(1) \tag{63}$$

$$E_{Case_6,r}^X(2) = (t_p^X + t_d) \cdot P_r + t_a^X \cdot P_t + E_r^X(1) \tag{64}$$

$$E_{Case_6,l}^X(2) = (t_l + (\gamma - 1) \cdot t_a^X) \cdot P_l + t_l \cdot P_l + \frac{t_p^X + t_a^X}{2} \cdot P_l + E_l^X(1) \tag{65}$$

$$E_{Case_3,s}^X(2) = (3 \cdot t_f - (t_l + \gamma^X \cdot (t_p^X + t_a^X) + t_d) + t_l + (\frac{t_p^X + t_a^X}{2} + t_p^X + t_a^X + t_d)) \cdot P_s + E_s^X(1) \tag{66}$$

$$E_{Case_6,o}^X(2) = E_{Case_3,o}^X(2) = 2 \cdot E_o^X(1) \tag{67}$$

– Case 7: The second transmitter wakes up as first, it hears a part of the strobed preamble until the receiver wakes up and sends an ACK. In average, when the second transmitter wakes up, it performs a short polling process whose duration is the one between two successive short preambles: $(t_p^X + t_a^X)/2$. After that, it hears an average number of $\lfloor \gamma^X/2 \rfloor$ short preambles before the receiver wakes up and stops the strobed preamble by sending an ACK. Probability of this scenario is $p_{Case_7} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2}$.

$$E_{Case_7,t}^X(2) = (\gamma^X \cdot t_p^X + t_d) \cdot P_t + t_a^X \cdot P_r + (\lfloor \frac{\gamma^X}{2} \rfloor \cdot t_p^X + t_a^X) \cdot P_r + t_d \cdot P_t \tag{68}$$

$$E_{Case_7,r}^X(2) = (t_p^X + t_d) \cdot P_r + t_a^X \cdot P_t + t_d \cdot P_r \tag{69}$$

$$E_{Case_7,l}^X(2) = (t_l + (\gamma^X - 1) \cdot t_a^X) \cdot P_l + ((\lfloor \frac{\gamma^X}{2} \rfloor - 1) \cdot t_a^X + \frac{t_p^X + t_a^X}{2}) \cdot P_l + \frac{t_p^X + t_a^X}{2} \cdot P_l \tag{70}$$

$$\begin{aligned} E_{Case_7,s}^X(2) = \; & (3 \cdot t_f - (t_l + \gamma^X \cdot (t_p^X + t_a^X) + t_d) - (\tfrac{t_p^X + t_a^X}{2} + \lfloor \tfrac{\gamma^X}{2} \rfloor \cdot (t_p^X + t_a^X) + t_d) \\ - \; & (\tfrac{t_p^X + t_a^X}{2} + t_p^X + t_a^X + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{71}$$

From the overhearers point of view, this case is equivalent to Cases 4 and 5.

$$E_{Case_7,o}^X(2) = E_{Case_4,o}^X(2) \tag{72}$$

- Case 8: There is only one sender that sends two messages in a row. This last scenario happens with a probability equal to $p_{Case_8} = 1/N$.

$$E_{Case_8,t}^X(2) = E_t^X(1) + t_d \cdot P_t \tag{73}$$

$$E^X_{Case_8,r}(2) = E^X_r(1) + t_d \cdot P_r \quad (74)$$

$$E^X_{Case_8,l}(2) = E^X_l(1) - t_d \cdot P_l \quad (75)$$

$$E^X_{Case_8,s}(2) = E^X_s(1) - t_d \cdot P_s \quad (76)$$

When the sender is unique, energy consumption of the overhearers can be assumed quite the same as the one in case of a global buffer with one packet to send ($B = 1$).

$$E^X_{Case_8,o}(2) = E^X_o(1) \quad (77)$$

The overall energy cost is the sum of the costs of each scenario, weighted by the probability of the scenario to happen (as showed in Figure 11):

$$E^X(2) = \sum_{i=1}^{8} p_{Case_i} \cdot E^X_{Case_i}(2) \quad (78)$$

LA-MAC ($B = 2$)

When global buffer contains more than one message, there can be one or several senders. In this section we deal with the case $B = 2$. Energy consumption $E^L(2)$ depends on the number of senders as well as on how wakeups are scheduled. All different combinations of wakeup instants with their probabilities are given in Figure 16. With probability $(N-1)/N$ there are two senders, otherwise there is a single sender. Cases 1-7 refer to situations in which two senders are involved, whereas case 8 refers to a scenario with one sender. We introduce now some probabilities that will be used in the remainder of this section. As previously defined, let $p = t_l/t_f$ be the probability of quasi-synchronisation between two devices; $q^L = (t_l - t^L_a)/t_f$ and $w^L = (t_l - 2 \cdot t^L_p - t^L_a)/t_f$ respectively the probabilities that the second transmitter can catch the early ACK of first transmission within its polling period and that the receiver can catch the preamble of the second transmitter before the end of its polling period. If none of the previous situations happen, the receiver will not be able to send an ACK to this second transmitter and there will not be a second transmission in the current frame. Assuming that the second transmitter has listened to the early ACK sent by the receiver for clearing the preamble of the first transmitter, the second transmitter knows when the *rendezvous* time has been scheduled for data transmission; then the second transmitter knows if it has enough time to send its preamble and waits for its early ACK before the *rendezvous* time.

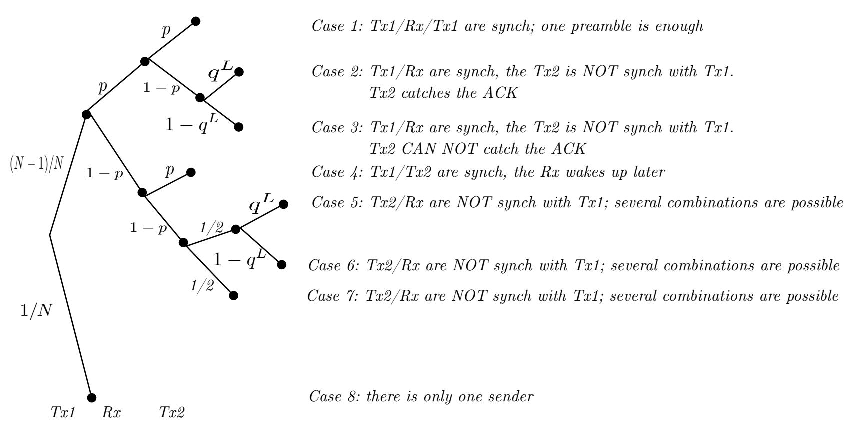

Figure 16. LA-MAC protocol, global buffer size $B = 2$: Probability tree of wakeup combinations.

- Case 1: The three active agents are quasi-synchronized. The very first preamble is instantly cleared by the receiver; the second transmitter hears this preamble and the ACK. This scenario happens with a probability equal to $p_{Case_1} = (N-1)/N \cdot p \cdot p$.
  Depending on the fact that the second transmitter succeeds or not in sending in time its preamble (probability w^L), there will be one or two frames requested for sending two data messages.

$$\begin{aligned} E^L_{Case_1,t}(2) = \;& (t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r \\ +\;& w^L \cdot (t^L_p \cdot (P_r + P_t) + 2 \cdot t^L_a \cdot P_r + t_g \cdot P_r + t_d \cdot P_t) \\ +\;& (1 - w^L) \cdot (t^L_p \cdot P_r + t^L_a \cdot P_r + E^L_t(1)) \end{aligned} \quad (79)$$

$$E^L_{Case_1,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + w_L \cdot (t^L_p \cdot P_r + t^L_a \cdot P_t + t_d \cdot P_r) + (1 - w^L) \cdot E^L_r(1) \tag{80}$$

$$E^L_{Case_1,l}(2) = (2 \cdot t_l - t^L_p - t^l_a) \cdot P_l + w^L \cdot (-(t^L_p + t^L_a) + \frac{t_l}{2}) \cdot P_l + (1 - w_l) \cdot (\frac{t_l}{2} \cdot P_l + E^L_l(1)) \tag{81}$$

$$\begin{aligned} E^L_{Case_1,s}(2) = &\quad (2 \cdot t_f - (t_l + t^L_p + t^L_a + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + &\quad w^L \cdot (-t_d + t_f - (\tfrac{t_l}{2} + 2 \cdot (t^L_p + t^L_a) + t_g + t_d)) \cdot P_s \\ + &\quad (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t^L_p + t^L_a)) \cdot P_s + E^L_s(1)) \end{aligned} \tag{82}$$

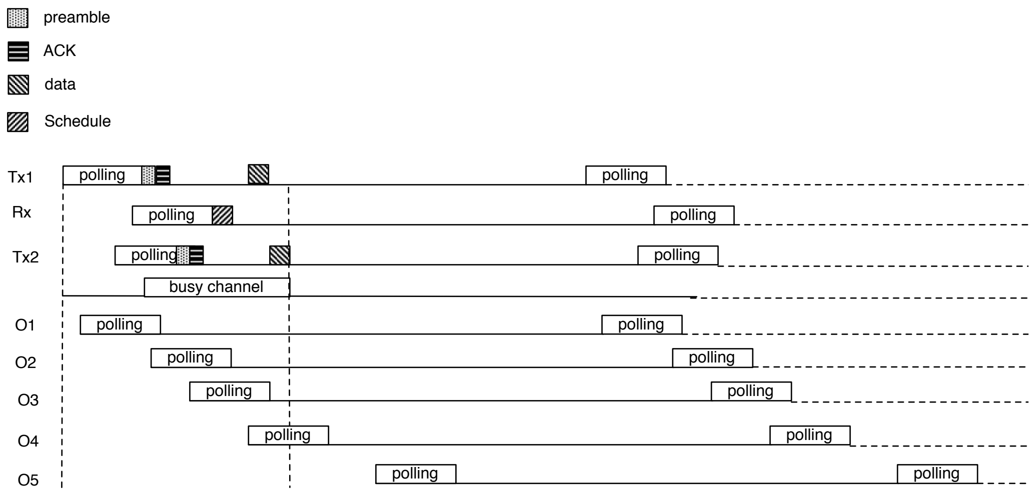

Figure 17. LA-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 1.

As far as overhearers are concerned, several situations may happen depending on their instants of wakeup. For simplicity we assume that if an overhearer is quasi-synchronized with one of the three active agents (sender one, sender two or the receiver), it will sense a busy channel (cf. Figure 17). We assume that the overhearer will poll the channel for some time and then ovehear a message (that can be a preamble, an ACK, a SCHEDULE or a data), for simplicity we assume that the ovehrearer polls the channel for an average time equal to half the duration (t_l) and then it will overhear a data (the largest message that can be sent). Probability to wakeup during a busy period is $p^L_{case1.1,B=2} = (2 \cdot (t^L_p + t^L_a + t_d) + t_g)/t_f$ if there are two data sent within one frame, otherwise probability is $p^L_{case1.2,B=2} = (t^L_p + t^L_a + t_d + t_g)/t_f$. Otherwise, if the overhearer wakes up while channel is free, it will poll the channel and then go to sleep.

$$\begin{aligned} E^L_{Case_1,o}(2) = &\quad N_o \cdot w^L \cdot p^L_{case1.1,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s) \\ + &\quad N_o \cdot w^L \cdot (1 - p^L_{case1.1,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s) \\ + &\quad N_o \cdot (1 - w^L) \cdot p^L_{case1.2,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s) \\ + &\quad N_o \cdot (1 - w^L) \cdot (1 - p^L_{case1.2,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s) \\ + &\quad (1 - w^L) \cdot E^L_o(1) \end{aligned} \tag{83}$$

- Case 2: The first transmitter and the receiver are quasi-synchronized. However the second sender, is not. The only possibility for it to send its data in the current frame is to send a preamble during the polling period of the receiver. Furthermore, the second sender must also catch its early ACK in time, this event happens with probability $q^L = (t_l - t^L_a)/t_f$. Nevertheless, even if the second sender catches the early ACK, the receiver must listen to the preamble of the second sender and still have time to send it its early ACK back (probability w^L).
  This scenario happens with probability $p_{Case_2} = (N-1)/N \cdot p \cdot (1-p) \cdot q^L$.

$$\begin{aligned} E^L_{Case_2,t}(2) = &\quad (t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r \\ + &\quad w^L \cdot ((t^L_p + t_d) \cdot P_t + (2 \cdot t^L_a + t_g) \cdot P_r) \\ + &\quad (1 - w^L) \cdot (t^L_a \cdot P_r + E^L_t(1)) \end{aligned} \tag{84}$$

$$E^L_{Case_2,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + w^L \cdot ((t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t) + (1 - w^L) \cdot E^L_r(1) \tag{85}$$

$$E^L_{Case_2,l}(2) = (2\cdot t_l - t_p^L - t_a^L)\cdot P_l + w^L\cdot(-(t_p^L + t_a^L) + \frac{t_p^L}{2})\cdot P_l + (1-w^L)\cdot(\frac{t_l}{2}\cdot P_l + E_l^L(1)) \tag{86}$$

$$\begin{aligned} E^L_{Case_2,s}(2) = & \quad (2\cdot t_f - (t_l + t_p^L + t_a^L + t_g + t_d) - (t_l + t_g + t_d))\cdot P_s \\ + & \quad w_l\cdot(-t_d + t_f - (\tfrac{t_p^L}{2} + t_p^L + 2\cdot t_a^L + t_g + t_d))\cdot P_s \\ + & \quad (1-w^L)\cdot((t_f - (\tfrac{t_l}{2} + t_a^L))\cdot P_s + E_s^L(1)) \end{aligned} \tag{87}$$

We assume that the probability of busy channel is the same as the previous case. So consumption is assumed to be the same as the previous case.

$$E^L_{Case_2,o}(2) = E^L_{Case_1,o}(2) \tag{88}$$

- Case 3: With probability $(1-q^L)$, the second sender wakes up too late and can not catch the acknowledge sent by the receiver to the first sender. In this case, the second sender will go back to sleep and will transmit its data during the next frame. Nevertheless, depending on the wakeup instant of the second sender, it can spend more or less time in each state. Let us define the remaining time $t_{remain} = (t_f - t_l/2 - t_p^L - t_a^L)$ as being the part of the receiver frame where the second sender can wake up. Let us also define $test = max(t_l/2 - t_p^L - t_a^L, 0)$ as being a variable that states the second sender wakes up just after the receiver has sent the early ACK (we just add a test of positivity).
This third scenario happens with probability $p_{Case_3} = (N-1)/N\cdot p\cdot(1-p)\cdot(1-q^L)$.

$$E^L_{Case_3,t}(2) = (t_p^L + t_d)\cdot P_t + (t_a^L + t_g)\cdot P_r + E_t^L(1) \tag{89}$$

$$E^L_{Case_3,r}(2) = (t_p^L + t_d)\cdot P_r + (t_a^L + t_g)\cdot P_t + E_r^L(1) + \frac{test}{t_{remain}}\cdot t_g\cdot P_r + \frac{t_g}{t_{remain}}\cdot t_d\cdot P_r \tag{90}$$

$$\begin{aligned} E^L_{Case_3,l}(2) = & \quad (2\cdot t_l - t_p^L - t_a^L)\cdot P_l + E_l^L(1) \\ + & \quad \tfrac{test}{t_{remain}}\cdot\tfrac{test}{2}\cdot P_l + \tfrac{t_g}{t_{remain}}\cdot\tfrac{t_g}{2}\cdot P_l + (1 - \tfrac{test + t_g}{t_{remain}})\cdot t_l\cdot P_l \end{aligned} \tag{91}$$

$$\begin{aligned} E^L_{Case_3,s}(2) = & \quad (2\cdot t_f - (t_l + t_p^L + t_a^L + t_g + t_d) - (t_l + t_g + t_d))\cdot P_s + E_s^L(1) \\ + & \quad \tfrac{test}{t_{remain}}\cdot(t_f - t_g)\cdot P_s + \tfrac{t_g}{t_{remain}}\cdot(t_f - t_d)\cdot P_s + (1 - \tfrac{test + t_g}{t_{remain}})\cdot(t_f - t_l)\cdot P_s \end{aligned} \tag{92}$$

Since there are two frames for sending the two data messages, the energy spent by overhearers is quite the same as the one detailed in previous section ($B = 1$).

$$E^L_{Case_3,o}(2) = \frac{N_o + N_o + 1}{N_o - 1}\cdot E_o^L(1) \tag{93}$$

- Case 4: First and second senders are quasi-synchronized but the receiver wakes up later. In this case, the first sender will send a strobed preamble and the second will hear all the preambles until the receiver wakes up and sends the ACK. This scenario happens with probability $p_{Case_4} = (N-1)/N\cdot(1-p)\cdot p$.

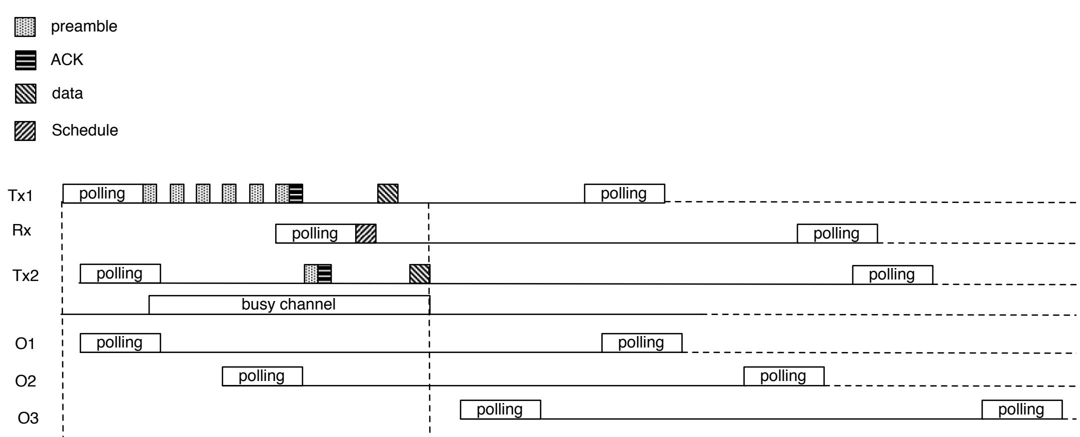

Figure 18. LA-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 4.

$$E^L_{Case_4,t}(2) = (\gamma^L\cdot t_p^L + t_d)\cdot P_t + (t_a^L + t_g)\cdot P_r + (t_p^L + t_d)\cdot P_t + (\gamma^L\cdot t_p^L + 2\cdot t_a^L + t_g)\cdot P_r \tag{94}$$

$$E^L_{Case_4,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + (t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t \quad (95)$$

$$E^L_{Case_4,l}(2) = (t_l + (\gamma^L - 1) \cdot t^L_a + t_l - t^L_p - t^L_a) \cdot P_l + (-(t^L_p + t^L_a) + \frac{t_l}{2} + (\gamma^L - 1) \cdot t^L_a) \cdot P_l \quad (96)$$

$$\begin{aligned} E^L_{Case_4,s}(2) = & \quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + & \quad (-t_d + t_f - \tfrac{t_l}{2} - (\gamma^L + 1) \cdot (t^L_p + t^L_a) - t_g - t_d) \cdot P_s \end{aligned} \quad (97)$$

If the receiver wakes up later than the couple of senders, the probability that an overhearer wakes up during a transmission of a preamble is high. If this happens, the overhearer performs a very short polling process, overhears a message (most probably a preamble) and then goes back to sleep. For simplicity, we assume the pessimistic case where the overhearer will perform half the duration of a polling period and then will overhear the longest type of message, *i.e.,* a data. Probability of busy channel is $p^L_{case4} = ((\gamma^L + 1) \cdot (t^L_p + t^L_a) + t_g + 2 \cdot t_d)/t_f$.

$$E^L_{Case_4,o}(2) = N_o \cdot (p^L_{case4} \cdot (\frac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_l}{2} - d) \cdot P_s) + (1 - p^L_{case4}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \quad (98)$$

- Cases 5, 6, 7: According to these three scenarios, the second transmitter and the receiver are not synchronized with the first transmitter; the behaviour of the protocol depends on which agent will wakes up as first among the second transmitter and the receiver.
  - Case 5: The receiver wakes up as first; similarly to Case 2, the only possibility for the second transmitter to send data in the current frame is to catch during its polling process the ACK of the receiver. This event happens with probability $q^L = (t_l - t^L_a)/t_f$. Nevertheless, the receiver is not quasi-synchronised with the first sender which will send a series of preambles before the receiver listens to one of them. This fifth scenario has a probability $p_{Case_5} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2 \cdot q^L$ to happen.

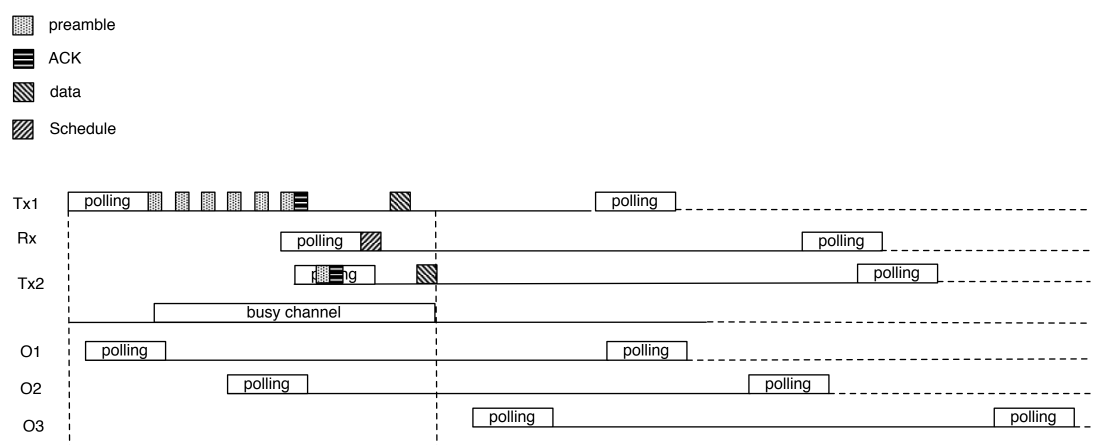

Figure 19. LA-MAC protocol, global buffer size $B = 2$: Overhearing situations for Case 5.

As explained previously, energy spent for the transmission of the second data message will depend on the probability of the receiver to catch in time the preamble sent by the second sender.

$$\begin{aligned} E^L_{Case_5,t}(2) = & \quad (\gamma^L \cdot t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r \\ + & \quad w^L \cdot ((t^L_p + t_d) \cdot P_t + (2 \cdot t^L_a + t_g) \cdot P_r) \\ + & \quad (1 - w^L) \cdot (t^L_a \cdot P_r + E^L_t(1)) \end{aligned} \quad (99)$$

$$E^L_{Case_5,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + w^L \cdot ((t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t) + (1 - w^L) \cdot E^L_r(1) \quad (100)$$

$$\begin{aligned} E^L_{Case_5,l}(2) = & \quad (t_l + (\gamma^L + 1) \cdot t^L_a + t_l - (t^L_p + t^L_a)) \cdot P_l \\ + & \quad w^L \cdot (-(t^L_p + t^L_a) + \tfrac{t^L_p}{2}) \cdot P_l \\ + & \quad (1 - w^L) \cdot (\tfrac{t_l}{2} \cdot P_l + E^L_l(1)) \end{aligned} \quad (101)$$

$$\begin{aligned} E^L_{Case_5,s}(2) = & \quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + & \quad w_l \cdot (-t_d + t_f - (\tfrac{t^L_p}{2} + t^L_p + 2 \cdot t^L_a + t_g + t_d)) \cdot P_s \\ + & \quad (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t^L_a)) \cdot P_s + E^L_s(1)) \end{aligned} \quad (102)$$

As in the previous scenario, the overhearer will perceive a very busy channel because of the transmission of preambles; it will wake up, perform half of t_l in polling state and than overhear a data.

$$E^L_{Case_5,o}(2) = E^L_{Case_4,o}(2) \tag{103}$$

– Case 6: The receiver wakes up as first, similarly to Case 3. With probability $(1-q^L)$, the second sender wakes up too late and can not catch the acknowledge. In this case it will go back to sleep and will transmit its data during the next frame. The first sender needs to send a series of preambles to wake up the receiver. Probability of this scenario to happen is given by $p_{Case_6} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2 \cdot (1-q^L)$.
The remaining part of the receiver frame during which the second sender can wake up, is given by t_{remain} as stated below. $test$ variable (see below) refers to the following case: "the second sender wakes up just after the transmission of early ACK".

$$\begin{aligned} t_{remain} = &\quad t_f - \tfrac{t_p^L + t_a^L}{2} - t_p^L - t_a^L \\ test = &\quad max(\tfrac{t_p^L + t_a^L}{2} - t_p^L - t_a^L, 0) \end{aligned} \tag{104}$$

$$E^L_{Case_6,t}(2) = (\gamma^L \cdot t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r + E_t^L(1) \tag{105}$$

$$E^L_{Case_6,r}(2) = (t_p^L + t_d) \cdot P_r + (t_a^L + t_g) \cdot P_t + E_r^L(1) + \frac{test}{t_{remain}} \cdot t_g \cdot P_r + \frac{t_g}{t_{remain}} \cdot t_d \cdot P_r \tag{106}$$

$$\begin{aligned} E^L_{Case_6,l}(2) = &\quad (t_l + (\gamma^L - 1) \cdot t_a^L + t_l - t_p^L - t_a^L) \cdot P_l + E_l^L(1) \\ + &\quad \tfrac{test}{t_{remain}} \cdot \tfrac{test}{2} \cdot P_l + \tfrac{t_g}{t_{remain}} \cdot \tfrac{t_g}{2} \cdot P_l + (1 - \tfrac{test + t_g}{t_{remain}}) \cdot t_l \cdot P_l \end{aligned} \tag{107}$$

$$\begin{aligned} E^L_{Case_6,s}(2) = &\quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t_p^L + t_a^L) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s + E_s^L(1) \\ + &\quad \tfrac{test}{t_{remain}} \cdot (t_f - t_g) \cdot P_s + \tfrac{t_g}{t_{remain}} \cdot (t_f - t_d) \cdot P_s + (1 - \tfrac{test + t_g}{t_{remain}}) \cdot (t_f - t_l) \cdot P_s \end{aligned} \tag{108}$$

For the overhearers point of view, this scenario is comparable to the one of Case 3.

$$E^L_{Case_6,o}(2) = E^L_{Case_3,o}(2) \tag{109}$$

– Case 7: The second transmitter wakes up before the receiver, so it will be ready to send a preamble immediately after the transmission of the ACK destined to the first transmitter. The second transmitter hears a part of the strobed preamble of the first transmitter: in average, it hears $\lfloor \gamma^L/2 \rfloor$ preambles. This scenario has a probability $p_{Case_7} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2$ to happen.

$$E^L_{Case_7,t}(2) = (\gamma^L \cdot t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r + (\lfloor \frac{\gamma^L}{2} \rfloor \cdot t_p^L + 2 \cdot t_a^L t_g) \cdot P_r + (t_p^L + t_d) \cdot P_t \tag{110}$$

$$E^L_{Case_7,r}(2) = (t_p^L + t_d) \cdot P_r + (t_a^L + t_g) \cdot P_t + (t_p^L + t_d) \cdot P_r + t_a^L \cdot P_t \tag{111}$$

$$\begin{aligned} E^L_{Case_7,l}(2) = &\quad (t_l + (\gamma^L - 1) \cdot t_a^L + t_l - t_p^L - t_a^L) \cdot P_l \\ + &\quad (-(t_p^L + t_a^L) + \tfrac{t_p^L + t_a^L}{2} + (\lfloor \tfrac{\gamma^L}{2} \rfloor - 1) \cdot t_a^L) \cdot P_l \end{aligned} \tag{112}$$

$$\begin{aligned} E^L_{Case_7,s}(2) = &\quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t_p^L + t_a^L) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + &\quad (-t_d + t_f - \tfrac{t_p^L + t_a^L}{2} - (\lfloor \tfrac{\gamma^L}{2} \rfloor + 1) \cdot (t_p^L + t_a^L) - t_g - t_d) \cdot P_s \end{aligned} \tag{113}$$

From the overhearers point of view, this case is equivalent to Case 4.

$$E^L_{Case_7,o}(2) = E^L_{Case_4,o}(2) \tag{114}$$

- Case 8: There is only one sender that will send two messages in a row. This last scenario may happen with a probability $p_{Case_8} = 1/N$.

$$E^L_{Case_8,t}(2) = E_t^L(1) + t_d \cdot P_t \tag{115}$$

$$E^L_{Case_8,r}(2) = E_r^L(1) + t_d \cdot P_r \tag{116}$$

$$E^L_{Case_8,l}(2) = E_l^L(1) \tag{117}$$

$$E^L_{Case_8,s}(2) = E_s^L(1) - 2 \cdot t_d \cdot P_s \tag{118}$$

When the sender is unique, overhearer consumption can be assumed the same as the case of $B = 1$. This is not exactly true since the busy activity of the channel is slightly larger, however differences can be neglected.

$$E^L_{Case_8,o}(2) = E_o^L(1) \tag{119}$$

The overall energy cost is the sum of the energy consumption of each case weighted by the probability of the case to happen (as showed on the Figure 16):

$$E^L(2) = \sum_{i=1}^{8} p_{Case_i} \cdot E^L_{Case_i} \tag{120}$$

D. Global buffer contains more than two messages ($B > 2$)

In the remainder of the document we seek to derive the generic expression for the energy consumption. Some hints can be extracted from the expressions detailed in the previous sections. We are more interested in deriving analytical results that give a relevant idea of the performance of each protocol rather than providing a tedious and complex study of all cases that may or may not happen.

B-MAC ($B > 2$)

With B-MAC protocol, energy consumption increases linearly with the number of messages in the global buffer independently on how packets are locally distributed, *i.e.,* independently of the number of senders. In fact, due to the long preamble to send ($t_p^b = t_f$), there can be only one sender per frame. Thus, we have the following relation: $E^B(B) = B \cdot E^B(1) = B \cdot (E_t^B(1) + E_r^B(1) + E_l^B(1) + E_s^B(1) + E_o^B(1))$.

Such a relation underlines the limitations of B-MAC protocol, since high-loaded traffic can hardly been addressed.

X-MAC ($B > 2$)

With X-MAC protocol, up to two messages can be delivered in each frame. After the transmission of the first data, any device with buffered messages to send can transmit without sending a preamble, if it wins the contention (back-off time). Nevertheless, the extra back-off time allows only one additional data per frame. If buffer size B is larger than 2, then necessarily at least two frames are needed. In the following expression we neglect collisions of preambles and messages so that the provided expression is *optimistic*. Without collision of preambles it results that frames are *efficiently filled*, that is, devices always use the minimal number of frames to send B data.

The computation of $E^X(B)$ is then quite straightforward and uses a modulo operator: if *B* is even we have just to compute the number of *full frames*, *i.e.,* frames during which two messages are sent; otherwise, if *B* is odd, we also add the cost for an extra frame for the remaining data. It follows the expression:

$$\begin{aligned} remain(B) =&\ rem(B, 2) \\ nb_{full\ frames}(B) =&\ \tfrac{B - remain(B)}{2} \\ E^X(B) =&\ nb_{full\ frames}(B) \cdot E^X(2) + remain(B) \cdot E^X(1) \end{aligned} \tag{121}$$

Consequently, the evolution of $E^X(B)$ with the increasing values of B is a step function, as depicted with Figure 21.

LA-MAC ($B > 2$)

With LA-MAC protocol, multiple senders can be scheduled per each frame. As we have done with X-MAC protocol in the previous section, we assume that the there are not collisions and that frames are *efficiently filled*, *i.e.,* each frame contains the maximum possible number of data.

The maximal number of data that a frame may contain is limited by either on the duration of a polling period and the duration of a frame (cf. Figure 20).

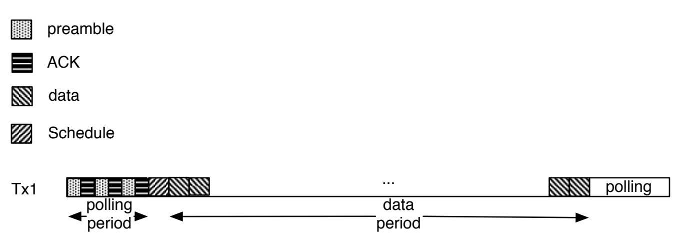

Figure 20. LA-MAC protocol, frame *efficiently filled* with data.

Thus two limitations are defined:

$$\begin{array}{rl} nb^{max}_{preambles} = & \lfloor \frac{t_l}{t^L_p + t^L_a} \rfloor \\ nb^{max}_{data} = & \lfloor \frac{t_f - t_l - t_g}{t_d} \rfloor \\ nb^{max}_{data\ per\ frame} = & min(nb^{max}_{preambles}, nb^{max}_{data}) \end{array} \tag{122}$$

To compute the number of necessary full frames as well as the number of data in the last and incomplete frame, we use a modulo operator:

$$\begin{array}{rl} remain(B) = & rem(B, nb^{max}_{data\ per\ frame}) \\ nb_{full\ frames}(B) = & \frac{B - remain(B)}{nb^{max}_{data\ per\ frame}} \end{array} \tag{123}$$

We provide a *pessimistic* analytical expression for LA-MAC energy consumption. The expression assumes that the energy consumed by all transmitters excepting the first one, is the same. So depending on the number of senders we multiply the unitary energy cost of transmission by the number of messages to send. The the unitary energy cost of transmission can be taken from the expression for $E^L(2)$ in the section corresponding to *B=2*. Let this amount be E^L_{tx2}; hence we have:

$$E^L_{tx1} = E^L(2) - E^L_{tx2} \tag{124}$$

In other words, we assume that E^L_{tx1} is a fixed part corresponding to the transmission of the first data and that E^L_{tx2} is then consumed for each additional data within this current frame. We have:

$$E^L_{pessimistic}(B) = nb_{full\ frames}(B) \cdot (E^L_{tx1} + (nb^{max}_{data\ per\ frame} - 1) \cdot E^L_{tx2}) + E_{last\ frame}(B) \tag{125}$$

where B is used to compute $nb_{full\ frames}$ and $remain$; besides, we also have for the last incomplete frame

$$\begin{array}{rll} E_{last\ frame}(B) = & remain(B) \cdot E^L_{tx1} & \text{if } (remain \leq 1) \\ = & E^L_{tx1} + (remain(B) - 1) \cdot E^L_{tx2} & \text{else} \end{array} \tag{126}$$

In this expression we assume that each transmitter has only one data message to send. With this basic assumption, we have to consider two possible cases:

- If $(nb^{max}_{data} < nb^{max}_{preambles})$, it means that the receiver will spend a part of its polling period without receiving any preamble. For this reason, in this case we set $nb^{max}_{data\ per\ frame} = nb^{max}_{data}$. and we assume

$$E^L(B) = E^L_{pessimistic}(B) \tag{127}$$

- Else, the receiver spends the entire polling period in receiving preambles and sending ACKs. Provided that the polling period has limited duration, also the number of preambles that can be sent during a single polling period is limited. In this case, we release the assumption that each sender has only one message to send: a full frame contains nb^{max}_{data} data distributed between $nb^{max}_{preambles}$ different senders. Thus, the polling period is efficiently filled (cf. Figure 20). Since $(nb^{max}_{data} \geq nb^{max}_{preambles})$, some transmitters will send more than one packet. We do not need to know how these data messages are distributed among all the $nb^{max}_{preambles}$ different senders.
  We derive below the complete energy consumption for a full frame. As previously mentioned, this energy is formed by the part E^L_{tx1} for the transmission of the first sender and by several times E^L_{tx2}. The total number of messages that are sent in a single frame is : nb^{max}_{data}. For each data message, sender and receiver spend a period t_d respectively in sending and receiving, instead of sleeping.

$$\begin{array}{l} \text{Number of data to send in the last frame:} \\ \quad remain(B) = rem(B, nb^{max}_{data}) \\ \text{Number of complete frames:} \\ \quad nb_{full\ frames}(B) = \frac{B - remain(B)}{nb^{max}_{data}} \end{array} \tag{128}$$

$$\begin{array}{rl} E^L_{full\ frame} = & (nb^{max}_{preambles} - 1) \cdot E^L_{tx1} + E^L_{tx2} \\ + & (nb^{max}_{data} - nb^{max}_{preambles} + 1) \cdot t_d \cdot (P_t + P_r - 2 \cdot P_s) \end{array} \tag{129}$$

  If the buffer size is larger than the maximum number of messages that can be sent in a single frame, we consider an additional frame. The last frame may be not complete, either because there are not enough senders to occupy the entire polling period, or because there are less than nb^{max}_{data} to send.

$$\begin{array}{lll} \text{if} & \multicolumn{2}{l}{(remain(B) \leq nb^{max}_{preambles})} \\ & \text{if} & (remain(B) \leq 1) \\ & & E^L_{last\ frame}(B) = remain(B) \cdot E^L_{tx1} \\ & \text{else} & \\ & & E^L_{last\ frame}(B) = E^L_{tx1} + (remain(B) - 1) \cdot E^L_{tx2} \\ \text{else} & & \\ & \multicolumn{2}{l}{E^L_{last\ frame}(B) = E^L_{full\ frame} - (nb^{max}_{data} - remain(B)) \cdot t_d \cdot (P_t + P_r - 2 \cdot P_s)} \end{array} \tag{130}$$

Finally, we can derive the overall energy consumption:

$$E^L(B) = nb_{full\ frames}(B) \cdot E^L_{full\ frame} + E^L_{last\ frame}(B) \tag{131}$$

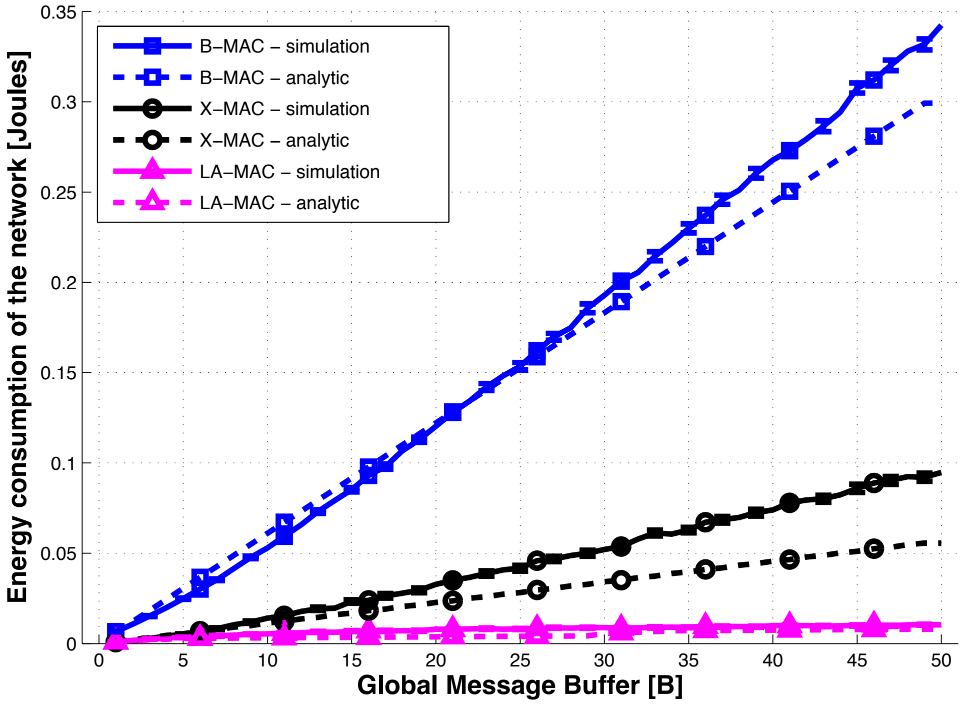

Figure 21. Energy analysis and OMNeT++ simulations versus the global buffer size.

V. Numerical Validation

We have implemented the analyzed MAC protocols in the OMNeT++ simulator [20] for numerical evaluation. Each numerical value is the average of 100 runs and we show the corresponding confidence intervals at 95% confidence level. We assume that devices use the CC1100 [21] radio stack with bitrate of 20Kbps. The values of power consumption for different radio states are specific to the CC1100 transceiver considering a 3V battery. In the following, we assume $N = 9$ senders. The periodical wakeup period is the same for all protocols: $t_f = t_l + t_s = 250\ ms$. Also the polling duration is the same for all protocols: $t_l = 25\ ms$, thus the duty cycle with no messages to send is 10%. We provide numerical and analytical results for buffer size $B \in [1, 50]$.

We compare the protocol performance with respect to several criteria:

- *Latency [s]:* the delay between the beginning of the simulation and the instant of packet reception at the sink (we present the latency averaged over all nodes).
- *Energy Consumption [Joules]:* the averaged energy consumed by all nodes due to radio activity.
- *Delivery Ratio:* the ratio of the number of received packet by the sink to the total number of packets sent.

In Figure 21, we show the comparison between the proposed energy consumption analysis and numerical simulations for different values of the global buffer size. We assume that at the beginning of each simulation all messages to send are already buffered. Each simulation stops when the last message in the buffer is received by the sink. Figure 21 highlights the validity of the analytical expressions for energy consumption—we can see that the curves reflect the main trends. The simulation results exceed the analytical data because the simulation reflects the detailed behavior for the protocols, which cannot be captured in simple expressions. As expected, B-MAC is the most energy consuming protocol: as the buffer size increases, the transmission of a long preamble locally saturates the network resulting in high energy consumption and latency (cf. Figure 23). In X-MAC, short preambles mitigate the effect of the increasing local traffic load, thus both latency and energy consumption are reduced with respect to B-MAC. Even if X-MAC is more energy efficient than B-MAC, Figure 22 shows that even for small buffer sizes, the delivery ratio for this protocol is lower than 100 % most likely because packets that are sent after the back-off collide at the receiver. LA-MAC is the most energy saving protocol and it also outperforms other protocols in terms of latency and the delivery ratio. We observe that when the instantaneous buffer size is lower than 8 messages, the cost of the SCHEDULE message is paid in terms of a higher latency with respect to X-MAC (cf. Figure 23); however, for larger buffer sizes the cost of the SCHEDULE transmission is compensated by a high number of delivered messages. In Figure 24, we show the percentage

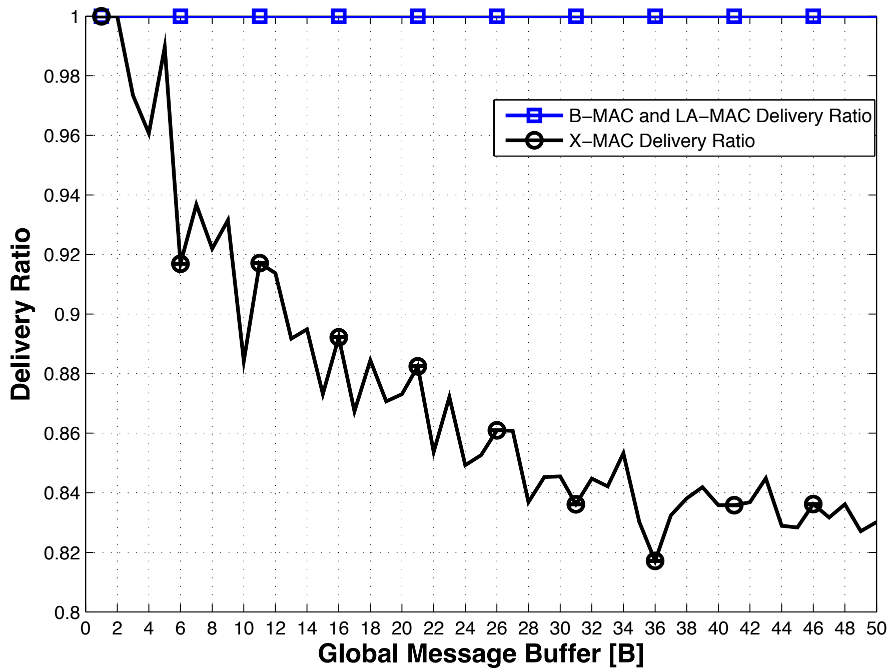

Figure 22. Delivery ratio versus the global message buffer. In X-MAC, most collisions happen when messages are sent after the back-off time.

of the time during which devices spend in each radio state versus the global buffer size. Thanks to efficient message scheduling of LA-MAC, devices sleep most of the time independently of the buffer size and all messages are delivered.

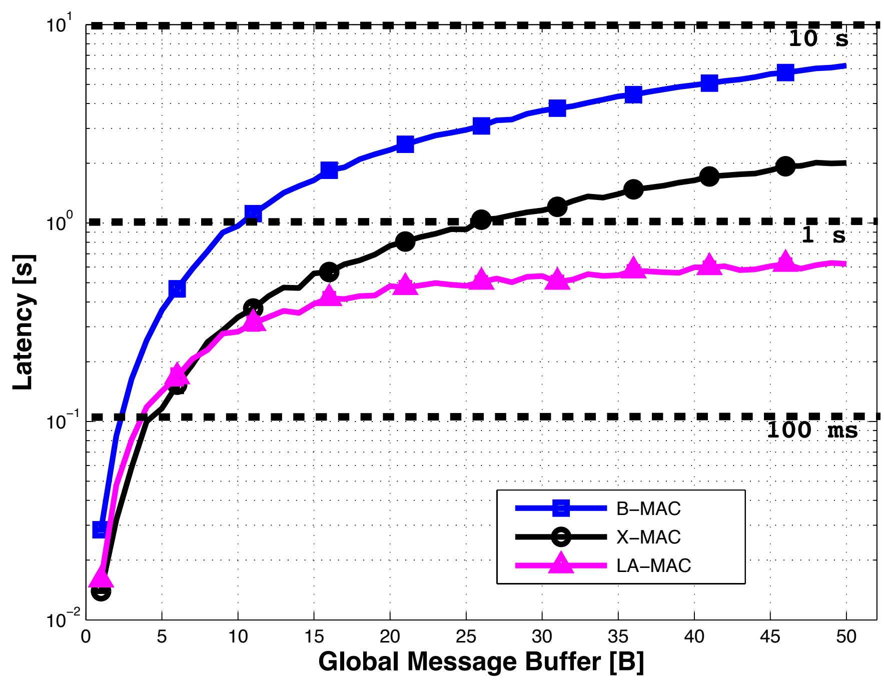

Figure 23. Average latency versus the global message buffer.

VI. Conclusions

In the present paper, we have analyzed the energy consumption of preamble sampling MAC protocols by means of a simple probabilistic modeling. The analytical results are then validated by simulations. We compare the classical MAC protocols (B-MAC and X-MAC) with LA-MAC, a method proposed in a companion paper. Our analysis highlights the energy savings

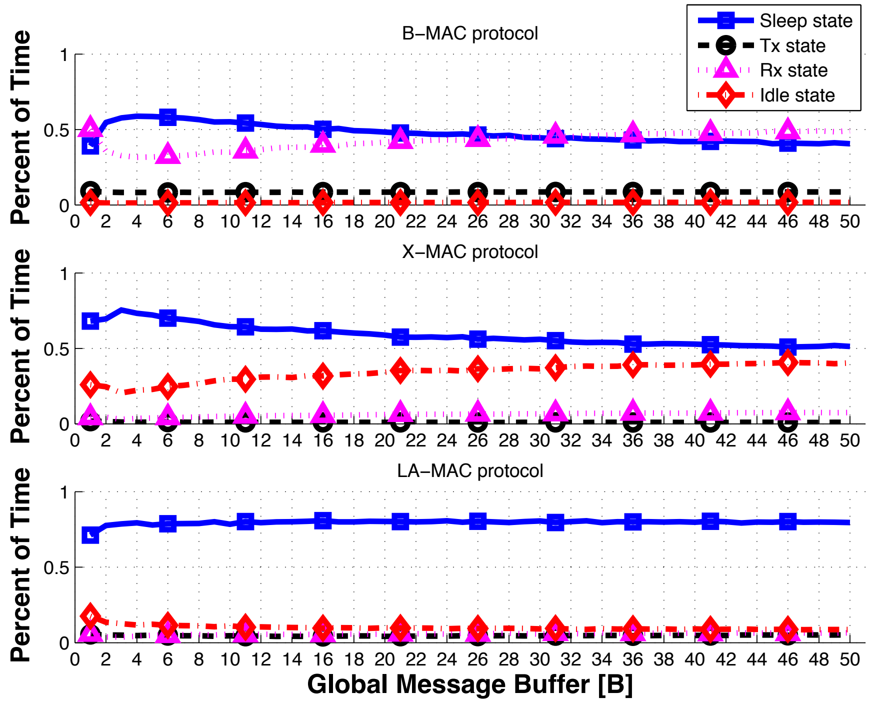

Figure 24. Percentage of the time spent in each radio state versus the global message buffer.

achievable with LA-MAC with respect to B-MAC and X-MAC. It also shows that LA-MAC provides the best performance in the considered case of high density networks under traffic congestion.

Acknowledgments

A part of this work has been performed in the framework of the ICT project ICT-5-258512 EXALTED partly funded by the European Commission. The authors Corbellini, Abgrall and Calvanese Strinati would like to acknowledge the contributions of their colleagues from the EXALTED project, although the expressed views are those of the authors and do not necessarily represent the project.

References

[1] "ICT-258512 EXALTED project." [Online]. Available: http://www.ict-exalted.eu/.

[2] K. Langendoen, "Energy-Efficient Medium Access Control," *Book chapter in Medium Access Control in Wireless Networks, H. Wu and Y. Pan (editors), Nova Science Publishers*, 2008.

[3] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *IEEE Infocom*, pp. 1567–76, New York, NY, July 2002.

[4] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *In Proceedings of ACM Sensys*, pp. 171–80, Los Angeles, CA, November 2003.

[5] L. van Hoesel, P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing Preamble Transmissions and Transceiver State Switches," *In Proceedings of IEEE INSS*, Tokyo, Japan 2004.

[6] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proc. of 18th IEEE IPDPS*, 2004, p. 224.

[7] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka, "Energy-Efficient, Application-Aware Medium Access for Sensor Networks," *IEEE MASS*, 2005.

[8] A. El-Hoiydi, "Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks," *IEEE ICC*, New York, NY, USA, April 2002.

[9] E-Y. Lin, J. Rabaey, A. Wolisz, "Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks," *In Proceedings of IEEE ICC*, Paris, France, June 2004.

[10] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *In Proceedings of ACM SenSys*, 2004.

[11] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the efficiency of preamble sampling protocols for wireless sensor networks," in *Proc. of the First Mobile Computing and Wireless Communication International Conference, MCWC*, 2006, pp. 117–122.

[12] S. Mahlknecht and M. Boeck, "CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks," *In Proceedings of IEEE Workshop on Factory Communication Systems*, Vienna, Austria, September 2004.

[13] M. Buettner et al., "X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Networks," *In Proceedings of ACM SenSys*, Boulder, CO, November 2006.

[14] R. Kuntz, A. Gallais, and T. Noel, "Auto-adaptive mac for energy-efficient burst transmissions in wireless sensor networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*. IEEE, pp. 233–238.

[15] G. Corbellini, E. Calvanese Strinati, E. Ben Hamida, and A. Duda, "DA-MAC: Density Aware MAC for Dynamic Wireless Sensor Networks," in *proceeding of 22nd IEEE International Symposium on PIMRC, Toronto, Canada*, September 2011.

[16] C. Enz, A. El-Hoiydi, J. Decotignie, V. Peiris., “WiseNET: An Ultralow-Power Wireless Sensor Network Solution,” *IEEE Computer*, vol. 37, no. 8, pp. 62–70, August 2004.
[17] W. Ye, F. Silva and J. Heidemann, “Ultra-Low Duty Cycle MAC with Scheduled Channel Polling,” *ACM SenSys*, Boulder, CO, USA, November 2006.
[18] G. Corbellini, E. Calvanese Strinati, and A. Duda, “LA-MAC: Low-Latency Asynchronous MAC for Wireless Sensor Networks,” in *submitted for publication*.
[19] C. Wan, S. Eisenman, A. Campbell, and J. Crowcroft, “Siphon: overload traffic management using multi-radio virtual sinks in sensor networks,” in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 116–129.
[20] “OMNeT++ Discrete Event Simulator.” [Online]. Available: http://www.omnetpp.org.
[21] “Texas Instruments, CC1100 datasheet.” [Online]. Available: http://focus.ti.com/docs/prod/folders/print/cc1100.html.